2011.10

PROGRAMMER 程序员



20 Windows 8 微软 BUILD 大会纪实

41 做好推广的核心是产品质量
Haypi 创始人任刚认为，没有好的产品质量，再完美的推荐最终结果都得不偿失

38 海豚浏览器 创业故事

62 论 Android 高效开发之路

89 如何使员工能力和收入相匹配

80 网络产品的内涵与产品机会评估

84 产品管理的前世今生

76 编程技术面试五大要点

94 MapReduce Hold 不住？浅谈流式计算系统

118 并行编程的曙光在哪？

ISSN 1672-3252

9 771672 325111 10

邮发代号：2-665　定价：15元

www.programmer.com.cn　www.csdn.net

Contents



资讯

8 资讯

10 新产品新工具

12 外刊速递

15 程序天下事

热点报道

20 Windows 8 大变局
微软BUILD大会于美国时间9月13日在加利福尼亚阿纳海姆会议中心召开，这是一场面向微软开发者的技术盛宴。全新Windows 8开发人员预览版的发布是本次大会焦点，它对于Windows的发展具有划时代的意义。

28 2011年分布式版本控制系统流行度大调查

30 勇敢者的变革
记第六届中国敏捷软件开发大会

32 软件让创新更智慧
记2011 IBM Rational创新论坛



封面报道
移动 ◆ 大航海时代

随着移动设备的崛起，Windows+Intel主导的PC帝国解体，世界正迎来信息产业的大航海时代。移动开放者们被财富、成功的前景和改变世界的热血所驱使，似乎忘记了远洋航行意味着冒险。巨大机遇面前，我们相信会有新一代的英雄横空出世，在新大陆雄起……

38 海豚浏览器的创业故事
海豚浏览器不仅在Android Market上大放异彩，被CNET评选为2011年十大最受用户欢迎Android应用之一，而且最近发布的iPhone和iPad版本，也很快冲到多个国家分类排名第一。本文讲述了其背后团队的创业故事。

41 做好推广的核心是产品质量
在Haypi创始人任刚看来，应用推广的核心是产品质量，没有好的产品质量，再完美的推荐最终结果都得不偿失。

44 《捕鱼达人》成功的另一面
《捕鱼达人》是触控科技旗下PunchBox团队开发的一款手机版深海捕鱼游戏。登陆App Store的首日就迅速攀升至付费应用榜的第一名。大量的技术积累、有效的市场推广是其成功的另类要素。

46 注重用户反馈迅速作出应对
InfoLife总部位于美国，其开发的应用在国内Android用户中有很高的知名度。近日，《程序员》采访了InfoLife团队，请他们分享创业经验。

Contents



48 移动互联网现在谈盈利还太早
从一无所有到拥有5000万的下载量，天天动听的开发团队已经取得了巨大的成功。而CEO黄晓杰却认为，在当前的移动互联网环境下谈盈利还太早。

50 移动互联网时代的语音识别技术
由科大讯飞公司开发的讯飞语音输入法自推出以来，已拥有数百万Android用户，其iOS版本也表现不俗，产品背后是科大讯飞国际领先的语音识别技术。

52 团结就是力量
RockPlayer创始人张永强认为，团队成员间的信任、团结和良好的相互沟通，造就了今天RockPlayer的良好口碑。

53 应用开发者如何善用广告盈利
移动广告已成为应用开发者主流盈利模式之一，Google商业合作部赵伊江指出，已有一批中国开发者通过AdMob年达到上千万元年收入，他建议应用开发者在产品设计之初就考虑到移动广告的设置和运营。

移动

56 用户体验质量控制体系
用户体验质量的评判要素有哪些？最容易出现哪方面的问题？本文为您讲解如何建立用户体验的质量控制体系。

58 手机社交游戏与触动用户的环节
手机社交游戏，一般的进程权衡可能会包含三个层面的考量：开发者的修为、发行商的布局和玩家的生态环境。本文为您解读其中触动用户最重要的环节。

62 论Android高效开发之路
本文通过对Android产业开发的简要分析，总结了Android系统开发中遇到的各种问题，从产品组合、需求管理、建模与重用、协同开发等角度分析和总结出能够加速Android系统开发并有效提高产品质量的方法和思路。

66 全景解读Android UI自动化测试框架TMTS（上）

P128



管理

70 对新事物要有好奇心
东软CTO张霞专访
1991年，几位教师仅凭3万元经费起家创立东软，现在的成绩有目共睹，其CTO张霞也几乎经历了东软20年的发展变迁。为此本刊对她进行了专访，希望通过她对东软产品技术战略的判断、理解，带大家一窥东软成功背后的秘密。

72 用户体验与敏捷开发的那些事儿

76 编程技术面试的五大要点
在作者看来，扎实的基础知识、高质量的代码、清晰的思路、优化代码的能力、优秀的综合能力是编程技术面试的五大要点。

TUP

80 网络产品的内涵与产品机会评估

2011年10月刊　总第228期

Contents

主管：中国社会科学院
主办：中国社会科学院文献信息中心
出版：《程序员》杂志社
网址：http://www.programmer.com.cn
国际刊号：ISSN 1672-3252
国内刊号：CN11-5038/G2
邮发代号：2-665
广告经营许可证号：京东工商广字0188号

总编：黄长著　Editor-in-chief：Huang Changzhu
社长/常务副总编：张悦校　President：Zhang Yuexiao
副社长：蒋 涛　Vice President：Jiang Tao
编委会：黄长著 张悦校 陈洋彬 蒋涛 曾登高 刘江
Editorial Member：Huang Changzhu Zhang Yuexiao Chen Yangbin
Jiang Tao Zeng Denggao Liu Jiang

执行主编：孟迎霞　Executive Editor-in-chief：Meng Yingxia
编辑部主任：董世晓　Director：Dong Shixiao
责任编辑：高松 陈博 陈秋歌
Editors：Gao Song Chen Bo Chen Qiuge
特邀编辑：方梁 高昂 赵健平 吕娜 卢鹄翔
Contributing Editors：Fang Liang Gao Aang Zhao Jianping
Lv Na Lu Dongxiang
美术设计：纪明超　Art Designer：Ji Mingchao
美术编辑：林象海　Art Editor：Lin Xianghai
流程编辑：白羽中　Coordinator：Bai Yuzhong
Tel：010-64351458
E-mail：editor@csdn.net

广告总代理：北京创新乐知信息技术有限公司
Sole Advertising Agency：Bejing CSDN Co.,Ltd
Tel：010-64376055
E-mail：ad@csdn.net
Marketing Dept：010-51661202（ext 149）
E-mail：market@csdn.net

发行部
Distribution Dept.010-64351431
E-mail：sales@csdn.net

读者服务部
Readers service Dept.
网上订购：http://dingyue.programmer.com.cn/
读者信箱：reader@csdn.net
地址：北京市朝阳区广顺北大街33号院1号楼福码大厦B座12层
Address: B-12th Floor Fairmont Tower NO.33 Guangshun North
street,Chaoyang District,Beijing
邮政编码：100102
电话：010-64351436
传真：010-64348545

法律顾问：北京中润律师事务所 王杰
Law Consultant：Beijng Hengsheng Lawyer Firm
印刷：北京盛通印刷股份有限公司
Print: Beijing Shengtong Printing Co., Ltd.
出版日期：每月1日
Publication Date：the first day per month
零售价：RMB 15.00元　新台币 390元　HK $ 35.00（港、澳）
US $ 9.00（海外）
Retail Price：RMB 15, NT$390, HK $ 35.00,US $ 9.00

本刊文章版权所有 未经许可不得转载
发现装订错误或缺页，请将杂志寄回本刊读者服务部调换

84 产品管理的前世今生——今天

一分钟先生

89 如何使员工能力和收入相匹配

云计算

94 MapReduce Hold不住？
浅谈流式计算系统

本文系统地介绍和分析比较了业界主流的Yahoo！ S4、StreamBase和Borealis三种流式计算系统，希望读者能从这些系统的设计中领悟到不同场景下流式计算所要解决的关键问题。

100 海量数据分析的“双立人”：YunTable

104 云计算新兴领域—安全即服务

技术

110 Redis源代码分析（下）
Redis是一个开源的Key—Value的内存数据库，以支持丰富的数据结构而著称，支持主从复制、持久化等高可用特性，可以和程序无缝地结合，本文继续上期分析Redis的源代码方法，帮助大家了解这一款数据库的工作机理。

115 快速响应的增量式开发
作者受到客户委托，开发一个应用程序，能够自动在拍卖中出价。他们勾画出它的工作方式和主要的组件。同时大致计划了增量式的步骤，来开发这个应用程序。

118 并行编程的曙光在哪？
如何在多核平台上有效地编程？本文分析了并行编程的前途与发展。

121 漫谈Android系统的调试模型（中）
本期将介绍Android系统的主机端（Host）的实现。

程序春秋

130 上善若水
Cordell Ratzlaff引发的Aqua革命（上）

名人堂

134 苹果锻造者

126 新书上架

128 Geek

136 程序幽默

移动互联网：颠覆还是延续



9月中旬访日四天，感触良多。日本上世纪八九十年代先后两次对国有独大的电信运营商NTT（日本电信电话公司）实现了民营化和上市、分拆重组，2001年更在全球第一个实现3G商用，为移动产业的发展打下坚实基础。

到今天，当地以功能手机为基础的移动产业已经非常成熟，生态链完善，除了几家大的社会化网站DeNA、Gree和Mixi都已经以移动用户和使用为主之外，还有不少移动互联网服务和游戏提供商都是上市公司，发展很好。以本次拜访的DeNA为例，2000多万用户，市值曾高达77亿美元，2010年收入13亿美元，主要是移动游戏所得，用户的ARPU值超过12美元，远远超过Facebook和Zynga这样的美国同行翘楚。日本移动平台上无论用户总数、总流量还是总收入，都已经超过PC。

相比之下，整个中国游戏市场的收入350亿人民币中，移动的份额只有百分之几，远不如DeNA一家。难怪访日同行的4399（这个小游戏网站有2亿注册用户）董事长蔡文胜感叹，在日本做游戏真幸福，有几十万、上百万用户就能活得非常滋润，而中国的游戏开发商必须将用户做到千万级才有可能赢利。

看上去很美的日本移动产业，其实正处在大变革的前夜。G-mode是日本老牌的移动游戏公司，成立于2000年，经历了日本移动平台从运营商门户转向DeNA、Gree和Mixi等社会化网站的变迁。在东京交流的时候，该公司的本田贵久分析，随着智能手机成主流，DeNA等网站也不再具有太大优势，有实力的各独立游戏商也都开始自己做平台。也许DeNA和Gree在自主游戏开发上都不遗余力，正是出于这样的原因。此外，Facebook和Twitter等外来平台也不容小视。金山公司负责日本业务的副总裁沈海寅就预测Facebook虽然目前市场份额尚小，但最终将席卷日本。

然而，Facebook在移动时代仍然能延续现在的优势吗？从商业模式上来说并非理所当然。Facebook的收入主要是广告和游戏、应用等收费分成，在移动平台上，广告会受影响，而游戏和应用平台更是无法简单移植。这正是Facebook要全力打造Project Spartan HTML5平台的原因。在中国现在令人畏惧的腾讯其实存在同样的问题，移动设备上QQ的威力显然已经大打折扣，也难怪马化腾眼下最重视的产品不再是微博，而是微信。但微信如何能在新世界里也成长为强大的平台，挑战也不小。

的确，在智能移动设备全新产业链里，出现了许多新兴力量。愤怒小鸟的开发商Rovio是最著名的代表。在国内有iOS上的Triniti、Android上的Droidhen等创业团队拥有了近亿级下载，也出现了捕鱼达人、Haypi这样优秀的游戏开发团队，和海豚浏览器、TalkBox等有平台潜力的应用。而在传统公司拥抱移动的大军中，网易新闻客户端在新战场对新浪、搜狐、腾讯等老对手取得了未曾有过的优势。

但是，这些已经和即将投入移动大变革的弄潮儿中，会出现下一个Google、Facebook、腾讯、百度，或者最终实质性改变产业格局吗？

访日期间，微软Build和Intel IDF两个会议几乎同时分别在安纳海姆和旧金山举行。微软发布了有全新界面Metro UI和全新底层技术平台WinRT的Windows 8，这也是首个支持ARM架构的Windows版本。Intel更是将Android之父Andy Rubin请为座上宾，高调宣布未来芯片要为Android优化。有意思的是，已经被业界很多观察家唱衰微软和Intel这次分别发布的产品都显得很有朝气，很有竞争力。

在新技术对企业影响的研究中，Clayton Christensen的颠覆性创新（disruptive innovation）理论最为人所熟知。他将新技术分为两类，延续性创新（更高性能、更多功能、更可靠）是原有企业增加市场份额的工具，而颠覆性创新（更简单、更便宜、更易用）才会赋予新进入者改变世界的利器。那么，在云计算大背景下的移动互联网，应该归属于哪一类呢？

让我们共同到本期移动团队的报道以及后续更多文章中，寻找答案。



邮件：liujiang@csdn.net
新浪微博：@刘江CE
欢迎大家交流反馈，欢迎投稿、批评、建议和挑错

Google再购IBM千余项专利

根据美国专利商标局（US Patent and Trademark Office）9月公布的信息显示，Google已从IBM手中购买了1023项专利技术，来支撑其防御智能手机诉讼的战略。这已是Google第二次购买IBM的专利。

Google当前正在打造一个专利库。该公司表示，此举的目的主要是针对苹果和微软等公司对Android操作系统发动的“敌对的、有组织的”专利诉讼战役。今年7月，Google已从IBM手中购买了1030项专利，并将通过125亿美元收购摩托罗拉移动获得超过1.7万项专利。

Google开发的Android平台是一款免费、开源系统，依赖于一些并非Google发明的非专利功能，并允许外部开发者修改代码。这使得Google和使用该操作系统的手机制造商容易遭到侵权诉讼，这些诉讼指控Android依赖于其他科技公司的研究成果。Android手机制造商宏达电、三星电子和摩托罗拉移动，此前都已成为苹果攻击的对象。与此同时，微软和摩托罗拉移动也在相互指责对方侵权。

为帮助Android手机制造商，Google于8月向宏达电转让了9项专利，这些均是Google在过去一年里从摩托罗拉和Openwave系统等公司收购获得的专利。宏达电使用这些专利在上周反诉苹果，使得两家公司的专利诉讼大战进一步升级。

惠普管理层隐瞒公司实情误导股东遭起诉


惠普新任CEO李艾科能否在错误的时间做出正确的抉择

惠普股东在9月的一起集体诉讼中称，惠普上月公布的一系列战略决定，例如放弃TouchPad，影响了该公司股价，然而惠普及其高管此前一直在误导投资者。惠普股东Richard Gammel表示，惠普此前隐瞒了该公司面临的实际情况，即该公司当前的业务模式表现不佳，而webOS系统也不是惠普业务模式的中心。8月18日，惠普宣布将考虑分拆PC业务，并放弃webOS硬件设备，例如TouchPad平板电脑的运营。此外，惠普还以120亿美元收购了英国软件公司Autonomy。在随后一个交易日中，惠普股价下跌了20%，创下自1987年美国股市崩盘以来惠普股价最大单日跌幅。

惠普股东此次委托Robbins Geller Rudman & Down律师事务所对惠普提起诉讼。诉讼称，包括CEO Leo Apotheker（李艾科）和CFOCathie Lesjak在内的惠普高管曾对公司业绩作出正面评价，这已被证明是没有依据的，因此误导了投资者。这起诉讼要求惠普对在2010年11月22日至2011年8月18日之间买入惠普股票的投资者进行损失赔偿，因为惠普没有充分披露信息，导致股价被人为抬高。惠普尚未对这起诉讼置评。

在所投资公司的股价大幅下跌之后，股东对公司提起集体诉讼很常见。目前，惠普股东对李艾科的不满正越来越明显。惠普近几个季度的业绩令人失望，一些股东还认为，惠普收购Autonomy的价格过高。

全新内核产品UC8亮相

9月18日，UC在北京宣布，部署了自主研发浏览器内核U3的新一代手机浏览器产品UC8正式亮相。UC是加入Chrome、Firefox等全球自有内核浏览器阵营，拥有全部内核能力的首个中国浏览器公司。同时基于U3内核提供的强劲动力，UC8在速度、智能、安全、低能耗、可扩展五项手机浏览器能力指标上都相比前代产品大幅提升，全面支持Web App，自此手机浏览器竞争进入核能力云服务能力综合比拼的时代。

必应搜索每季度巨亏10亿美元

微软必应搜索引擎已经上线两年，每个季度的亏损接近10亿美元，而且没有任何停止亏损的迹象。自从在2009年6月发布必应以来，微软已经在该业务上亏损了55亿美元。不过微软的搜索亏损实际上早已开始，事实上，微软在线服务部门仍未实现盈利，自2007年以来已经亏损了90亿美元。根据互联网监测机构comScore发布的数据，必应现在的市场搜索份额为14.7%，高于发布时的8.4%。谷歌现在的份额为64.8%。

“我们没有任何补贴，但我们要提供服务的方便，同时公司也有一些套餐配套的措施。”

——中国移动董事长王建宙在2011年夏季达沃斯论坛期间透露，截至2011年8月底，移动iPhone在网用户已达到853万，且7月增长速度超百万。但移动和苹果尚未就引入iPhone签署协议。

“Facebook、Twitter、LinkedIn普及程度高，小型、新推出社交网络面临巨大挑战。”

——2011年9月15日市场研究公司InSites Consulting在调查全球35个国家9027名15岁以上消费者之后发布了《2011年全球社交媒体报告》。

“我希望我和Steve Jobs之间没有闹翻，我希望能回到当初，对他说‘这是你的公司，让我们想想办法，你回去出任CEO’，但遗憾的是，历史是无法更改的。”

——苹果前CEO John Sculley在接受采访时表示，Steve Jobs当时被赶出苹果公司是董事会的决定，他本人不应为此承担责任。

“我觉得，现在是时候离开Mozilla了。”

——Mozilla联合创始人、副总裁Mike Shaver宣布从Mozilla离职，他在Mozilla工作六年，曾担任过包括工程副总裁在内的多个职位。

“如果仅仅考虑未来一年，那么纯HTML和Flash可以实现的工作仍会有很大不同。Windows桌面仍将是Windows非常重要的一部分，因此在未来几年中，Flash仍将是互联网体验的重要部分。”

——微软表示Windows 8的“Metro”风格IE10浏览器将不支持Adobe的Flash插件。Adobe副总裁Danny Winokur如此回应。

“可以确定我们处于后PC时代。我们处在一个全新的领域中，很难预计未来会发生什么。我认为，移动设备的发展推动了后PC时代的到来。”

——Sun Microsystems创始人、风险投资公司Khosla Ventures合伙人Vinod Khosla在TechCrunch Disrupt大会上接受专访，他谈论“后PC时代”行业的发展时如是说。

“不会对外出售。”

——雅虎董事会辞退了首席执行官Carol Bartz之后，刚被任命为雅虎临时首席执行官的Tim Morse，雅虎董事会主席Roy Bostock、雅虎联合创始人杨致远、David Filo以及公司其他高管出席了1.35万雅虎员工参加的全员大会。杨致远在会议中对雅虎的未来表态。

CSDN十大新闻

2011年8月26日~9月26日

01 知名程序员云风辞职

作者 / 子夜

网易杭州研究中心总监云风（吴云洋）9月离职，引来业界广泛关注。他在博客中表示，“想先全身心的放松”，将投入攀岩活动。云风是国内知名的程序员，网易主打游戏《大话西游》、《梦幻西游》、《网易泡泡游戏》等网络游戏的主力开发者。

02 IT人员迅速提升效率的十大方法

作者 / Wayne Harrel

Zielix创始人、Ingiter Frameworks创建者Wayne Harrel发表的博文。包括远离灰色地带；制订一个结果驱动型的计划；雇用一个顶级优秀的教练；寻找一款高性能的软件，并尝试应用它；形成具有说服力的个人想法；把你做该事情的激情和动机写下来并行动起来等。

03 十大科技公司创始人离职后发展回顾

作者 / Matt Rosoff

失去一名具有远见卓识的创始人并不是公司的末日，但这些公司通常会伴随着创始人的离开而转变命运。文章回顾了十家科技企业在创始人离职后的发展状况，多数企业逐渐陷入困境，甚至最终消亡。其中包括我们熟知的Steve Jobs、Bill Gates、杨致远等人。

04 程序员应知道的12件事

作者 / Robert Diana

“怎样成为优秀的程序员？”作者强调优秀程序员应该关注的开发语言及要了解的算法和其他技能。此外，优秀程序员的生活还包含哪些方面？作者分享了自己做程序员的12个经验心得，从选择喜欢的饮料到掌握睡眠时间等，非常实用。

05 微软2011 BUILD大会

作者 / 孟迎霞

9月13日，微软BUILD大会在美国开幕。大会主角是全新Windows 8开发人员预览版以及安装了Windows 8的平板电脑。微软多位高管登台演讲，为开发人员分享了Windows 8的用户体验特色、新的应用模型/开发平台和工具、安全与企业功能、云应用等。

06 这一天终于来了：乔布斯辞职

作者 / 刘江

8月24日，乔布斯宣布辞去CEO职务，转任公司董事会主席。在乔布斯的强烈建议下，原COO Tim Cook被提名接任CEO，并加入董事会。董事会声明中称，乔布斯杰出的远见和领导挽救了Apple，并带领公司成为世界最具创新力和最有价值的技术公司。



07 马云给职场新人立规矩

作者 / 徐洁云、王如晨

在一封写给入职新人的邮件中，马云称“刚来公司不到一年的人，千万别给我写战略报告，千万别瞎提阿里巴巴发展大计，谁提，谁离开！”外界视此举动为马云因一连串侵蚀阿里价值观以及马云个人道德光环的事件所采取的修补举措。

08 开发者应关注的Visual Studio 11新特性

作者 / 夏梦竹

微软将把Visual Studio运行在Azure上，将SaaS提供给开发者。其新增特性包括：为Windows 8开发Metro风格应用程序；增强（2D/3D）游戏开发；使用Team Explorer进行代码审查；Exploratory Testing和Unit Testing；快速创建错误报告；管理探索式测试。

09 英特尔谈未来Ultrabooks和Windows 8

作者 / John Callaham

在花旗技术大会的演示中，英特尔CFO Stacy Smith展示了一款类似于平板电脑款式的Ultrabook。新的设计会给笔记本电脑带来新的发展空间，Smith对此表示：“现存的笔记本领域在过去的几年中发展有所停滞，Ultrabook带来的新元素将改变笔记本市场的原有模式。”

10 Facebook遭遇7大问题：多项服务失败

作者 / Matt Rosoff

Facebook开发人员大会预计将发布一项新的音乐服务。此外，还可能推出新的移动支付平台、iPad应用或iPhone图片共享应用。但文章仍然列出了Facebook所遭遇的诸多问题，其中包括Places、Deals、Messages、Groups和Pages等数项服务失败；iPhone应用升级慢等。

2011全球最具价值IT品牌排行榜

9月，英国品牌价值咨询公司Brand Finance公布了2011年全球最具价值品牌排行榜，排名前三位的企业分别是Google、Apple和Microsoft。Brand Finance数据显示，Google的品牌价值为482.78亿美元，同比增加9%，排名首位。Apple的品牌价值为393.01亿美元，同比增加33%，位居第二。Microsoft的品牌价值为390.05亿美元，同比下滑9%，排名第三。

01 Google ↑

超过零售业巨人沃尔玛成为全球最有价值品牌，持续在线服务领域的统治地位，全球最知名的搜索引擎。

02 Apple ↑

发布了一系列富有革命性的产品，因此获得极大的成功，总是富有市场远见，拥有大批品牌忠实度极高的拥趸。

03 Microsoft ↑

以Windows7出色的销售业绩继续领跑个人电脑，Xbox已经成为最流行的游戏终端。Windows Phone 7有望在移动领域崛起。

04 IBM →

因一些先锋性的科技发明而闻名，在2011年迎来百年诞辰。其先进的服务在快速发展的新兴市场仍然大有可为。

05 HP ↓

世界最大的个人电脑制造商，也是数据存储、网络软硬件提供商。但品牌价值大幅缩水，打击了投资者信心。

06 Intel ↑

在移动、企业数字、家庭数字、医疗等领域均有所建树。在芯片、技术开发、产品与平台等领域全球领先。

07 Amazon ↑

美国最大的电子商务公司，在技术、商业上均表现不俗。公司旗下包括Alexa Internet、a9.com、IMDB等网站。

08 Oracle ↓

涉及服务器及其工具和应用软件两大领域，软硬件结合方面成绩卓著，在数据库、虚拟化、Java平台等方面全球领先。

09 Dell ↓

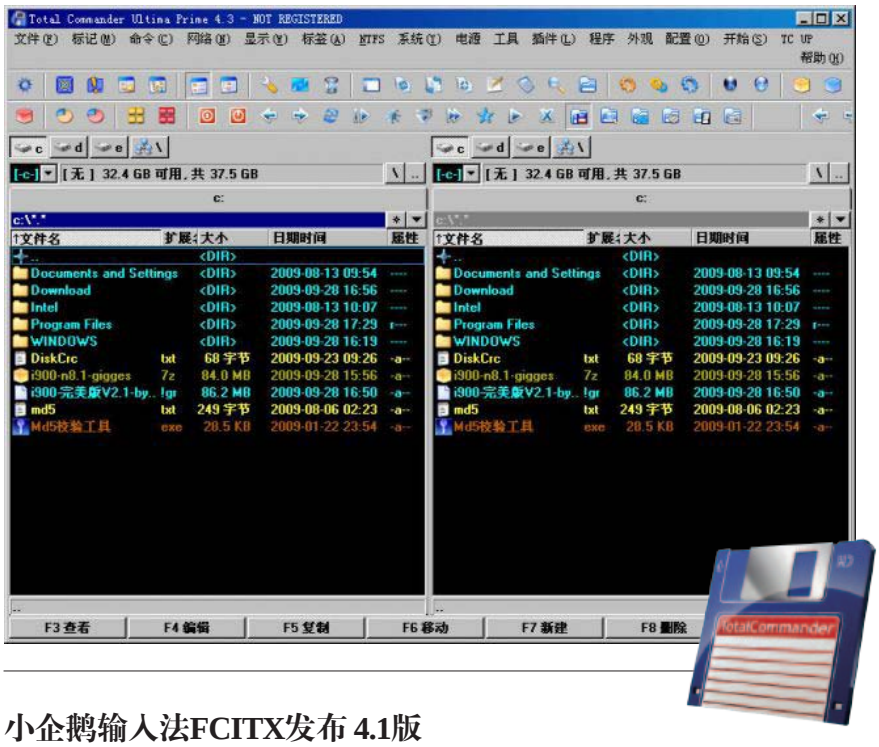
提供电脑及其相关产品和服务的企业，同时涉足高端电脑市场，服务器、数据存储设备、网络设备等。

10 SAP ↓

目前全世界排名第一的ERP软件，提供相关的企业级解决方案，还提供与其软件相关的咨询、维护和培训服务。

Total Commander发布

Total Commander是一款著名的磁盘文件管理软件，体积小巧但功能强大，可以完全取代资源管理器。除了具有一般的文件管理功能外，还能实现文件比较、快速搜索、快速查看、批量重命名、压缩、解压、分割、合并、编码、解码，FTP连接以及图像浏览等功能。Total Commander 8.0是第一款同时发布32位和64位的版本，但64位版本的插件较少。新版本修正了大量Bug也增加了许多新特性，如64位版本执行文件名是totalcm64.exe，可以与32位版本安装在同一文件夹等。



小企鹅输入法FCITX发布 4.1版

小企鹅输入法FCITX发布4.1版，增加Android上的Google拼音移植；增加OpenCC做简繁转换引擎支持；增加支援Fcitx所有拼音引擎的云拼音支持；默认界面修改。支持竖排候选词列表，改善绘制性能；增加与KDE整合的配置模块；增加GTK2、GTK3、QT4的输入模块，以期解决Firefox、GTK3、Flash等程序的输入问题；增加对fbterm的支持；采用doxygen及docbook完善部分开发相关的文档。

Zentyal 2.2 正式版发布

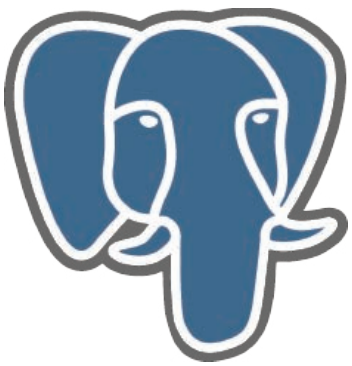
Zentyal是一份集成的网络服务器，它面向中小型企业提供简易和有效的计算机网络管理。它可以当作网关、基础设施管理工作站、风险管理工作站、办公服务器、通信服务器或是上述的集成来使用。这些功能被紧密地集成以自动完成大多数任务，并且避免出错和节省系统管理时间。Zentyal遵循GNU公用许可证（GPL）发布，它运行于Ubuntu之上。新版本进行了一些功能完善和Bug修补。

微软发布了Python编程开发工具

微软发布了Python编程开发工具Python Tools for Visual Studio（PTVS）。PTVS是一个开源项目，采用Apache 2.0许可发布。PTVS的主要特性包括：CPython、IronPython、Jython和PyPy；高级编辑功能如IntelliSense；多重构；内置REPL（read-eval-print loop）窗口；调试和分析功能等。

Infinispan 5.0.1.发布

Infinispan 5.0.1.FINAL发布了5.1.0的第一个Alpha测试版本，代号为Brahma。Infinispan 5.1.0包含以下主要新特性：完全支持分布式查询，可在所有节点中并行查询，只需要简单调用 SearchManager.getClusteredQuery即；Infnispan查询模块使用Hibernate Search 4扩展了执行操作的缓存处理。



开源数据库项目PostgreSQL发布9.1版

开源数据库项目PostgreSQL发布了v9.1版。新版本主要的特性包括：同步复制、序列化快照隔离、支持基于列的排序、近邻邻近索引、外来数据封装、支持SELinux许可控制等。

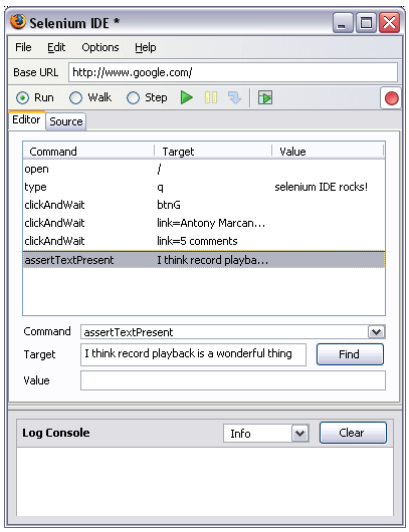
RabbitMQ 2.6.1 发布

RabbitMQ 2.6.1发布了，该版本修复了2.6.00版本中发现的Bug，这些Bug导致在某些操作系统上无法重启。同时该版本提供了第一个官方的可视化程序拓扑逻辑显示。

RabbitMQ是由LShift提供的一个 Advanced Message Queuing Protocol（AMQP）的开源实现，由以高性能、健壮以及可伸缩性出名的Erlang写成，因此也是继承了这些优点。

Selenium 2.6 发布

Selenium 2.6发布了，该版本主要包含一些提升和稳定性方面的改进。Selenium（SeleniumHQ）是ThoughtWorks公司的一个集成测试的强大工具。



Cosmos：基于C#的操作系统

一个使用C#编写的操作系统Cosmos已经发布了Beta版、Cosmos采用BSD开源许可证，包含了一个AOT编译器IL2CPU负责把.NET平台上的中间语言bytecode翻译成机器指令，开发者可以使用Visual Studio Express Edition编译成为可引导（bootable）二进制文件，可以通过软盘、USB设备（U盘）、CD-ROM和通过PXE来网络启动，或者安装到虚拟机中。社区计划未来会加入VB.NET的支持。

Grails 2.0 M2 发布，高效的Web开发框架

Grails 2.0发布第2个里程碑版。该版本的新特性：当使用@Mixin时，@Validateable不工作。

支持在默认排序中使用嵌套属性（例如：mapping {sort 'nested.property'}）。

改进包括：将默认的URL映射修改为对用户友好的URL，比如控制器和动作名称中的连字符<g:resource>可被配置；修复了当不使用g:datePicker时，从参数中检索日期比较困难的问题；支持公共和私有视图，比如Servlet规范；新增domain类auto-complete到create-scaffold-controller中；改进了GSP内容类型设置；修复了HibernateQuery无法处理StreamCharBuffer参数的问题；提供了一个“Refresh Depencies”命令选项，以下载源码附件。

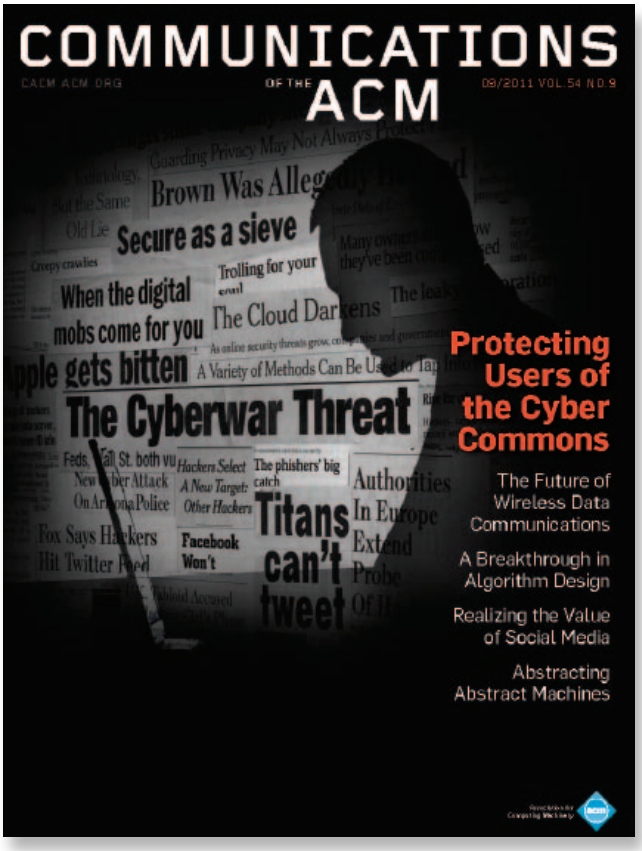
OpenSearchServer 1.2.3 RC2发布



OpenSearchServer 1.2.3发布了第二个RC版本，主要是Bug修复，另外新增了一个Drupal模块；在API方面和Web扒取方面也有一些改进；支持sitemap生成和搜索结果导出CSV文件格式。OSS（Open Search Server）是一个用Java开发的搜索引擎服务器，基于Lucene、ZK和Tomcat等开源软件架构。

IPFire 2.9 Core 52 发布

IPFire是一份Linux发行，它注重轻松的装备、方便的操作和高级别的安全。它通过一份直观的基于网页的界面来进行操作管理，该界面为新手级及老练的系统管理员提供很多直观的配置选项。IPFire由一群关注安全及经常更新该产品以保持其安全的开发者来维护。新版本进行了一些功能完善和稳定性方面的提高。



Communications of ACM

2011.09

算法设计的突破

线性系统的应用非常广泛。电信、交通、制造业以及许多其他领域都不乏它们的身影。针对线性系统的算法必须能够求解包含数百万甚至数亿个变量的方程组。即使在最快的计算机上，这些计算方案都十分费时，因此，寻找一种加快计算速度的方法是算法设计面临的一个长期挑战。如今，卡内基•梅隆大学的一组计算机科学家设计出了一种算法，可以求解一类特定的线性系统，其速度远远快过如今的求解器。这种算法适用于对称对角占优（SDD）系统。SDD系统的用途广泛，从在线推荐系统到工业模拟，从材料建模到图像处理，都有它的用武之地。这种类型的线性系统算法分为两大类：直接求解和迭代求解。直接求解方法需要消耗大量的内存；因此对于如今实际应用中产生的大量数据集而言，迭代求解方法更加有效。虽然迭代求解最终能返回令人满意的结果，但这些结果通常需要较长的时间。为了加快迭代求解速度，目前最有效的方法是通过快速计算与原系统在代数意义上类似的稀疏系统，来求解决大规模线性系统。而预处理器被用于建立原系统的稀疏系统。

2004年，Dan Spielman和Shang-Hua Teng证明了，对于SDD系统，存在着一个最优的预处理器。他们设计的斯皮尔曼-腾求解器的速度远远大于之前的求解器，而且随着系统规模的增大，该求解器在速度上的优势更加明显。但他们的算法只是在理论上可行，而且是针对“巨大得不切实际”的系统矩阵。卡内基•梅隆大学的研究团队在此基础上开发的算法不仅在理论上可靠，而且具有实用的潜力，速度相当快，一台普通台式机在几秒钟之内就能求解有着10亿个变量的系统。如果再加上算法实现中的优化，对同样的问题，该算法将比其他求解器快10~20倍。

移动应用的入侵

据市场研究公司Gartner预测，今年移动应用程序商店的下载次数将达到177亿次，比去年的82亿同比增长了116%。而预计到2014年，应用程序下载次数将飙升至1850亿次。

据其统计，2011年，开发人员从移动在线应用程序中获得了超过15亿美元的收入，包括下载费用和链接广告。成千上万的小型创业公司借助苹果、谷歌和其他几个公司的在线应用程序商店，绕开了传统的开发、营销、分发模式，降低了经营成本。得益于这种新的商业模式，软件创业公司Spring partner的客户数量在14个月内飙升了3900%。他们开发的资料收集应用Springpad同时在苹果和谷歌的在线应用商店上架。Springpad的所有版本都是免费的，其收入来源于用户对存储数据采取的行动，比如购买图书。虽然在线商店拿走了一部分利润；但对软件开发人员来说，这是一种免费的分销机制。该公司的一个业务重点是保证其应用在商店中获得尽可能高的排名。

另外一家只有两个人的创业公司“Instant Cocoa”也受益匪浅，其创始人之一Maland使用苹果的SDK，用一星期时间创建了一个简单的SSH客户端应用pTerm。应用上架的第一天就有4000余人下载，现在他每月能卖出1500~2000份拷贝，每份4.99美元。另一个创始人Block开发的益智游戏2 Across也获得了差不多的成绩。软件开发能如此快速的原因之一，是用户并不要求文档、帮助菜单，以及售后支持。而这种新兴的商业模式也对那些销售现成软件，并且开发周期动辄数年的消费类软件公司提出了警告。

Nuance公司已经有几十年的历史，在应用程序商店存在之前，他们通过零售商和分销商，销售其语音产品“Dragon”。但现在Nuance公司已经开始为苹果移动设备提供产品——其最有名的“Dragon”听写和“Dragon”搜索应用。Nuance公司最近也开始发布核心的语音识别软件的SDK和API，使任何开发人员可以很容易地在产品中整合Nuance语音产品。

无线数据通信的未来

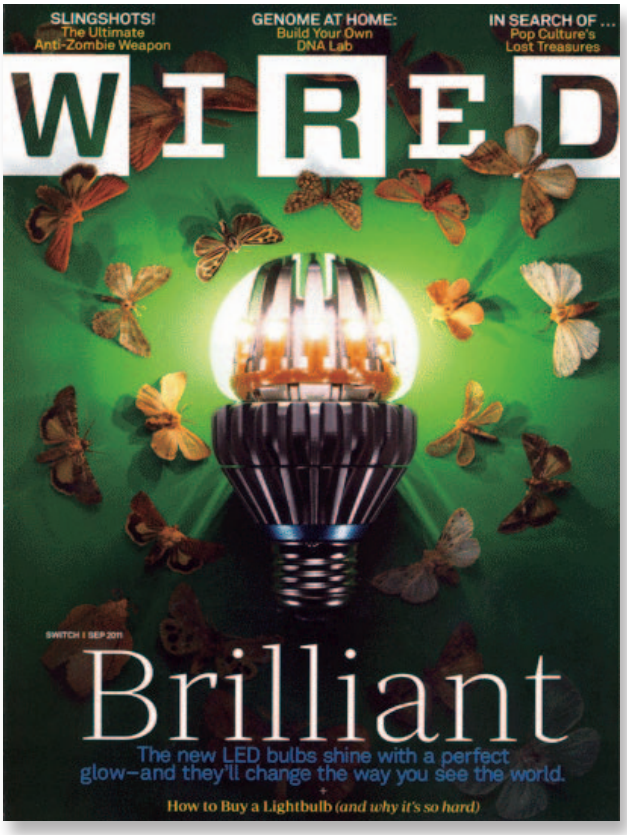
本文中，作者大概描述

了无线数据通信的技术路径。到2020年左右，软件无线电将成为标准技术。在软件无线电中，所有或几乎所有的功能，从频率和编码的物理层（如PSK - QAM），到媒体接入层（如时分复用和载波感测多重存取），都可以通过无线电上运行的软件实时改变。软件无线电带来了无与伦比的灵活性，这种灵活性来自电台的行为是由软件决定的事实。此外，软件控制行为改变了创新的步伐。在传统的硬件世界中，无线电行为的改变，要等待6个月或以上。

而在软件世界中，变化能马上发生，只需一个程序员的连夜编译和调试。作者设想了几种实现软件无线电的未来世界的途径。一种是建立一个标准的广播频道，不断转递适合某一特定区域的软件；另一种方法为每个使用中的协议分配一个标准组织注册名，在协议数量不多，且假定每个无线电预加载了所有协议的软件（或配置信息）时，这是最好的方法；再一种方案是分配给每个地理区域一个无线网络管理者，告知设备当地领域内的无线电规则；第四种方法是，在无线电中内置认知推理模块。无线电定期扫描可用频率的频谱，并推理出相应的规则。未来的无线环境会带来很多机遇。一个最明显的变化是，人们购买设备时，不用再考虑它采用什么协议。同一个设备，在不同的环境中，可以自动改变通信协议，必要时，还可以下载新的通信协议，而这一切都不需要用户的干预。另一个变化是带宽，当应用程序需要更多带宽时，软件无线电可以寻找未使用的频率，并向基站请求该频率，以获得所需的带宽。

无线电工程师已经建立了软件无线电平台。监管机构也开始提供频谱供这些无线电试验。但还有很多重要问题有待研究。最迫切的需要是研究描述无线电行为的语言。软件工程师需要某种途径在现场向不同的无线电描述某个协议，好让它们可以立即运行该协议。如何让软件无线电适当使用频谱，也是一个值得关注的问题。另外，政府研究机构应该资助无线电芯片组的开发。

很多具有挑战性的问题，在无线电芯片组上解决起来会更容易。研究人员还需要对可用频谱进行更多的测量，以便更好地了解在世界范围内有多少频谱是被广泛使用的。此外，我们需要了解未充分利用的频谱有多少可用带宽。如果这些研究能在未来五年内完成，我们就能有足够的信息作出明智的选择。



Wired

2011.09

Groupon：Web 2.0的背叛者？

和每一代人有自己的特点一样，互联网的每一波创新都有自己的特色。亚马逊、eBay、雅虎是互联网中最伟大的一代，它们创立了一种新的民主电子商务的形式。还有谷歌，它使我们可以接触到所有的人类知识。Web 2.0混合了真实世界的社交联系，它产生于新世纪这一代对技术的偏爱以及渴望互联的本性。互联网上出现的最新潮流就是团购：与本地商家联合提供产品优惠。本文作者Steven Levy认为这一潮流看起来更像是历史的倒退，没有创新的感觉，倒像是报纸优惠券的网络升级版。

这一趋势的领潮人是总部在芝加哥的Groupon，这家有3年历史的公司由31岁的安德鲁•梅森创立。梅森的初衷是创办一家以团体为单位的、关于慈善捐款的社交网络公司，而Groupon仅仅是他创业失败的副产品而已。2010年年底，Groupon拒绝了谷歌60亿美元的收购要约，目前它正在急切地等待公司的首次公开募股（IPO）。Groupon的模式就是和本地商家合作，向消费者提供折扣，去年它获得了7.6亿美元的营收，但也因为没有利润而饱受批评。与此同时，模仿

Groupon的网站如雨后春笋般冒出。现有的互联网巨头也觊觎本地市场，并积极开发自己的系统。在被Groupon拒绝后，谷歌开始开发自己的团购项目Offers，目前正在测试之中。Facebook也通过Deals项目进入了这一领域。

Steven Levy认为团购折扣模式缺乏此前互联网运动的革命品质。Groupon等公司并没有充分利用数字经济的优势，而是在自助模式下依靠少量雇员使自己发展壮大。折扣设定是劳动密集型的工作。需要销售人员在电话上向商家解释，为什么这个模式会奏效，并说服他们以新增客户为目的，在几乎无利润的情况下售卖商品。然后还需要文字工作者给这些平凡的商品附上华丽辞藻。把重点放在折扣上，对于一家公司来说并没有什么错，特别是如果这样可以帮助本地商家拓展业务。Steven Levy的批评主要来自于这种现象代表了理想主义的减弱，而正是这种理想主义让互联网变得特别，让世界发生巨大改变，让人们可以快速获取知识，让失散多年的朋友还能重逢。

网络广告的问题

许多年来，流传着这样一个说法，你不应该为在线服务收费。如果你开发了一款很酷的在线工具，获得收益的唯一途径是放置广告，并尽可能地为广告创造更多关注。这就不可避免地产生了可怕的产品设计，与用户的需求背道而驰。以Facebook为例：每一年，该网站都会进行重新设计，以逐步推动用户公开更多的个人资料。这种做法，部分是因为其CEO马克•扎克伯格似乎认为公共性在本质上是好的，但也是对广告市场要求的一种理性反应，因为广告市场需要尽可能多的人，以及尽可能多的需求。

相比之下，收费服务让你的需求与用户的需求保持一致。对这些付钱给你的人，你会感到一种神奇的关系，你会在乎他们的需求。另外一点是，付费服务提供商不需要成群的商务开发人员，可以专注于解决技术问题，而不是整天想办法让用户点击广告。

Pinboard.in是Ceglowski公司的一个书签服务，它允许用户管理链接，查看有哪些人保存了它们，甚至可以缓存整个网页的副本。它只有20000名用户，与Delicious，一个历史悠久的书签网站是竞争对手，它们有许多相同点。但有一个最大的不同：Delicious一直采取免费策略，而Ceglowski公司却对Pinboard收取大约9美元的入门费，而如果想使用网站的缓存副本功能，要每年支付25美元。

表面上看来，这似乎是一种自杀行为。为什么当你庞大的竞争对手免费时，你却要收费？Ceglowski的回答只有一个词：广告。由于用户付给报酬，他并不需要投放广告，然后从客户的注意力中赚钱。他不用去做那些通常会毁掉用户体验的事情，比如塞满网页的让人分心的横幅或把一整段内容分解成几小块，增加用户点击。如果不直接收费，你就只能做这些违背道德的东西，出售他们的注意力。

事实证明，Ceglowski并不孤单。新一代的网络企业家已经发现收费用户的好处。提供“稍后阅读”服务的Instapaper，私人社交网络站点Ning，和无数的iPhone应用程序都要求用户付费。

如果我们足够幸运，这一趋势将终结互联网最有腐蚀性的影响因素——广告。

埃里•克里斯的创业公司新理论

你脑中有关创业公司的一切都是错误的。埃里•克里斯在他的新书《精益创业》（*The Lean Startup*）中，表达了这样一个观点。他的观点受到了硅谷的权威，如Tim O'Reilly、马克•安德森和米奇•卡普尔的支持。《连线》就他的新书对他进行了访谈。

访谈中，他表示对于创业公司，人们最常见的误解是：相信只要天时地利，再加上正确的观念，创业就一定能够成功。但没有实际的证据，证明这是真实的。而实际上，有成千上万的证据证明它并不代表真正的实际情况。举例来说，20世纪初，在亨利•福特进入汽车行业时，还有很多其他的企业家进入了这一领域：当时有大约500家汽车公司得到了资助，他们有内部的动力，有技术，有远见。但其中有60%的公司在一两年内就关门大吉了。也许有人认为这是因为亨利•福特有他的竞争对手所没有的远见卓识。但是，如果你看过福特的纪录片，你会得到不同的结论。他不清楚将会发生什么，但福特公司能够适应境况的变化。所以关键在于，在过程中，能随时调整自己的视野，或者调整整个公司。

为了实现这一点，克里斯个人的做法是：持续部署。人们通常认为，行动慢一点会更好，因为可以及时修正错误。但事实并非如此。行动越慢，就会有更多更严重的问题出现。如果客户并不需要你的产品呢？不应该在整个产品或是一部分产品完成的时候，才发现这个问题。发布不完整或者糟糕的产品不是一个错误，错误是发布了这样的产品，而且还让公关人员参与其中。不应该让人们对产品的早期版本有过于兴奋的预期。最好的创业环境是低风险的环境，是没有人关注时，就像扎克伯格在哈佛大学的宿舍里那样。也就是说，开始的时候，只拥有少量客户是更好的。这意味着你可以非常好地了解客户，并且没有成功的压力。你可以慢慢花时间搞清楚哪些做法是有效的，然后再一击制胜。

（感谢译者卞斌支持）

Pad市场群雄逐鹿

平板电脑市场未来将会如何？本文做了9点预测，可以看出竞争更加激烈，产品更加多元，将是必然的趋势。

移动市场风云变幻，曾经一家独大的iPad越来越受到来自方方面面的竞争者的威胁。就在9月初，微软发布了新的Windows 8操作系统，还有安装了Windows 8的几款Pad产品，正式加入了Pad市场的竞争。

说起Pad，不得不提到早期的Tablet PC，中文同样叫作平板电脑，而后者正是微软早在2000年就最先提出的理念，其特点是用户的输入通过触摸屏而并非键盘和鼠标来进行操控，不过运行的还是经过裁剪的桌面操作系统。那时由于触摸屏技术所限，还都是早期的单点手触摸屏或电阻屏，而且运行的也是未专门为Pad优化过的软件，效果可想而知。除此以外，10年前，移动网络对于人们来说还只是工作偶尔用之，网络速度也完全不能和现在相比，不用说浏览视频，就连一般的浏览网页也是一件很痛苦的事情。所以虽然那时的Wintel联盟主推了一阵儿，但还是不被市场接受，后来便逐渐销声匿迹。

直到2010年，苹果借着iPhone的成功以及移动互联网的逐渐成熟，在当时成熟的软硬件平台上发布了iPad。iPad具有与iPhone相同的操作系统iOS，同样灵敏的多点触摸电容屏幕，同样精心设计过的应用和游戏，支持WiFi和3G两种联网方式。一经推出，就受到了人们的狂热追捧，短短80天内卖出了300万台，而当年更是卖出了高达1400万台。说iPad重新发明了平板电脑，并且让Pad这个词进入了移动电子产品的中心，真是一点都不为过。

Pad市场火了，自然就有竞争对手加入，在2011年初的消费电子展上，有超过80款新的平板产品发布：摩托罗拉的Xoom（Android 3.0）、三星的Galaxy Tab（Android 2.2/3.0）、RIM的Playbook、HP的基于webOS的TouchPad，以及众多采用了Android 2.2或者3.0的产品。苹果随后的对策是推出了iPad 2，有更高的配置和更强的性能，但价格与iPad保持一致，并把iPad一代的价格降低。到目前为止，苹果还是最大的赢家，iPad 2进一步畅销。而再回头看看之前的竞争对手们：不用提基于Android 2.x系列的众多小Pad，就是基于专门为Pad产品开发的Android 3.0的众多产品，大家都很有期待，可惜其从推出就问题不断，兼容性、稳定性、应用数量都有问题。只有三星的Galaxy Tab销售比较可观，其他产品都情况不佳。尤其是呼声最高的Xoom，受其高高在上的价格拖累，被苹果的价格战打得销售量惨不忍睹。再来看RIM的Playbook，多花了几个月时间修复了耗电问题后，于4月中旬上市了，而结果如何呢？到目前为止，仅仅销售了20万台，就在近日，RIM还计划降价促销。这些还不是最惨的，最惨的是当初号称iPad 2最大竞争对手的TouchPad，尽管其价格从上市时的499.99美元一降再降，到399.99美元，但还是只销售出了其计划数不到10%，最终导致HP决定停止webOS相关的消费电子产品，包括TouchPad。不过值得讽刺的是，在决定终止产品并且把TouchPad以99美元的价格清仓甩卖，一下子造成了TouchPad的供不应求，看来还是价格才是真正的决定因素。

TouchPad的悲剧发生不到一个月，微软就通过Windows 8的发布会宣布正式加入Pad产品的竞争者行列。我还是比较看好微软这款新系统在Pad上的表现。在之前的分析中，我就说过如果将Windows Phone 7平台移植到Pad上，将会非常合适，并且也预测了微软会推出基于Windows Phone平台的Pad产品，时至今日，算是基本应验了。Windows 8虽然不能说是Windows Phone 7的移植产品，但也可以说是延续了Windows Phone 7的主要特性，尤其在Pad的定制OS上，和Windows Phone 7的关系非常紧密。可以预计，将来Windows Phone 7和Windows 8也会在移动OS领域融合成为同一套系统，就如同iOS和Android要做的一样。对于微软这位昔日的王者、曾经的平板电脑先驱，此次联合x86和ARM两大阵营一起高调回归，尽管最终结果如何还未可知，但我们起码可以大胆预测一下未来的平板电脑市场。

- iPad/Android/Windows三分天下，其中Android将会占据半壁江山，iPad次之。
- 平板电脑硬件配置将会越来越高，明年将会不少平板安装4核CPU和4G内存，到了2013年，你要不装个双核CPU，都不好意思拿出来给别人看。
- 平板电脑屏幕大小将会从5~11寸不等，价格差距也因此分成高、中、低三等。
- 会有越来越多的企业逐渐淘汰普通笔记本和台式机办

公，取而代之的是平板电脑。

■ 只具有单一功能的类平板电脑产品，如视频播放器、彩色阅读器等产品会被平板产品取而代之。

■ 在电池技术获得突破性进展前，平板电脑的CPU应该不会超过8核，或者有这种产品应该也不会有太多人问津，制约移动智能设备产品发展的还是在于电池技术。

■ 应用为王，没有足够的开发者支持的Pad产品终将退出市场。

■ 微软将会在Windows 9上将Windows Phone 7和Windows 8合为一体，以一个系统打天下，同时支持电话、平板、PC三大平台。

■ 在解决了电池问题后，会有头戴式、手表式、纸张式等超便携设备参与移动产品的竞争。

未来实际情况是怎样我们现在不得而知，不过可以确定的是一定比现在精彩。



崔海斌
十几年的开发经验，专注于Java、Android、移动开发等领域。现于创新工场旗下的点心团队就任总架构师。

PostgreSQL的应用生命周期管理

如今MySQL被甲骨文把控，另一个开源数据库PostgreSQL正在逐步成为开源数据库的旗手。

对于用惯了Visual Studio Team System和Rational Software Architect的用户而言，数据库应用的开发生命周期管理已经是司空见惯的事情，从需求、设计、开发、测试一直到上线部署都可以在开发平台的统一管理下以“可追溯的”方式“水到渠成”。但对于开源数据库的开发人员，情况远没有这么美好，从开发过渡到测试、从测试部署到运行环境往往需要在字符界面下面堆一大堆的脚本和命令，即便是被招安的MySQL，尽管设计、建模、测试能力有了很大改进，但在从测试到运行环境部署的过程中仍然要面对大量的手工操作。对于其他的开源数据库，其开发生命周期的工具支持较之MySQL更是不及。

上面的情况如果只是针对几个数据库项目，也许不是问题，毕竟DBA的工作很多时候拼的就是“手熟”，评价一个DBA经验多寡一定程度与他多年积攒的“私房脚本”的质量和数量不无关系，但如果面对的是一个庞大的基于开源数据库的公有云环境，这就有点勉为其难了。近期，商业开源数据库厂商EnterpriseDB和虚拟化技术公司VMware瞄准这个需求，分别通过各自的产品对PostgreSQL进行了完善：

■ 针对PostgreSQL难于部署安装、难于监控的问题，EnterpriseDB发布了PostgreSQL企业管理器（PEM：PostgreSQL Enterprise Manager），这个工具不仅界面非常“企业级”，而且能管理的内容也非常“企业级”，一改以往PostgreSQL界面生涩刻板的面貌，而且能对实机实例（Instance）和虚拟机实例实现统一管理界面，同时对于常规的诊断、分析也可以在同一个的图形化界面中完成。

■ VMware则针对数据库从开发到部署的过程发布了新产品vFabric Data Director，通过vFabric技术开发团队可以根据定制的模板，快速面向不同开发需求发布用于开发和测试的虚拟机环境，而且可以将通过测试的虚拟机环境通过工具直接克隆到生产环境，而以前这个切换过程往往意味着一定的硬件设备占用、一系列开发部门和运维部门的沟通会议、一摞摞部署文档和一批必须通过配置管理管控的数据库脚本。虚拟机的想法虽然不新鲜，但在数据库应用生命周期过程的切换中却很实用。

在被SUN作为“嫁妆”嫁入甲骨文之后，尽管MySQL依然开源，但在独立开发团队中似乎味道已经不同于以往，因此PostgreSQL乘势而上，逐步成为开源数据库旗手也就可期了。不过同是“开源”，因为以前MySQL的风头过盛，以至于发展到2009年尽管PostgreSQL在开源数据库圈子中的排名已经很靠前，但整体比例还是太低。但自从MySQL有了新“东家”之后，一部分目光开始从MySQL和NoSQL逐步转向PostgreSQL，毕竟关系数据库在事务处理类应用中还是比较合适的，加之PostgreSQL已经具备事务、子查询、数据完整性检查等常规功能，因此逐步受到

重视。随着“云”概念的流行，PostgreSQL数据库也开始被批量部署到分布式环境，这时候工具支持不足的问题就越发明显。不过，有需求就不愁有产品，EnterpriseDB和VMware的新产品覆盖了从开发一直到部署运维的生命周期过程，但还是缺少从需求到设计的工具，不过这个关系不大，因为开源社区已经有一批备选的产品，而以前只是没有把这些需求、设计工具具体落地到PostgreSQL头上，相信EnterpriseDB会在后续版本中把这“最后一公里”修通。

当今，一个数据库是否有配套的生命周期管理工具的支持，关系到它能否被广泛接纳并在各种商业案例中获得不断改进的机会，“开源≠简陋和粗制滥造”，EnterpriseDB和VMware的两个新产品正好帮助PostgreSQL迈出了这关键的一步。



王翔
软件架构师，主要研究方向为XML、.NET、领域设计和PKI应用。工作之余喜爱旅游、写作和烹饪。

江湖紊乱，该如何选择？

软件行业风云突变，无论移动互联，还是企业软件变革得让人摸不到头脑，如何创新，如何选择呢？

PC时代之王，Microsoft和Intel，面对iPhone和iPad的步步紧逼，也都分别作出了自己的选择。ARM版Windows 8平板华丽亮相，英特尔CEO Paul Otellini也在9月14日的IDF大会上作出承诺——明年将有数家手机厂商推出基于x86芯片的Android手机。Wintel联盟还是那个Wintel联盟，只不过被限制在PC领域，PC也被限制在工作和特定的领域，而伴随千万人娱乐功能的设备，已经悄然从PC脱离，进而进化为带有移动功能的iPad和iPhone。一个月前，我换了一部新手机，没等作为手机用，却被我孩子拿来当作娱乐设备，天天在线看《花园宝宝》。而以前高价买回来的台式机、上网本、笔记本就闲置在那里，不再使用了。所以，风投极度关注移动互联网，还是有道理的。

Wintel联盟虽没有解体，但面对更大的原来几十倍甚至上百倍的娱乐加移动的市场，作为软件开发企业和软件程序员的我们，该做出如何的改变和选择呢？前些天还和同事讨论，国内很多中间件生产商，也有不少是海归归来，苦苦挣扎十年了，声音却越来越小。而中关村一个攒机器出身的刘强东，却已忽悠了十几亿美元的风投，超越Google，即将成为互联网历史上最大的IPO。仅仅是行业的不同，却造成以后的发展如此的不同，就像一个孩子的出生，有的出生在英国皇家，有的出生在伊拉克和利比亚，孩子没有过错，错就错在出生在了不同的地点和不同的家庭。软件行业也是如此，选错了行业，像一个美丽的姑娘嫁错郎一样，都是要遗憾终生的。难道我们就这样自行解体，转向电子商务了吗？

当然，对于软件企业，好消息也不是没有的，最近一个月，云服务基础设施软件系统供应商6fusion，获得B轮融资700万美元，目前累计融资1000万美元，投资由风险投资公司Grotech Ventures所主导。9月6日，云服务平台创业公司Morphlabs，通过云服务平台通过PaaS模式为用户提供云服务，获500万美元第三轮投资，累计高达1200万美元融资。其他还有SaaS软件商Assistly发布AppExchange应用，融合Salesforce客户关系管理等功能，2011年初，Assistly公司获得新一轮300万美元风投。前SAP高管等成立的人力资源分析类商务软件Visier创业公司获600万美元融资。可惜这些商业融资，都是发生在国外。至于国内，VC们听到云计算就头疼了，更别说融资了。而国家的冤枉钱，其实也没少花，可惜国家投入的这些钱，最终不知道花到哪里去了，都无声无息地流失了。

虽然融资拿到钱，并不能说明什么，但靠做项目养活团队的日子，同样也是很艰难的。面对互联网背景下的移动大潮，能做的创新，还是很多的。

刚才说了，选错了行业，我们将经历无法摆脱的痛苦，但选错了开发语言的话，同样会令我们懊悔不已。这

不，Microsoft在最近的BUILD大会上对开发者说，.NET、WPF、Siverlight统统是浮云，Metro才是你们的未来……好消息是Silverlight和.NET不会寿终正寝，坏消息就是它们主要只能用来开发传统/桌面应用了，因为Metro应用才是Windows 8的主流。不要说新鸟发愁学什么、用什么开发语言，即使像我这样的老鸟，也在为选择什么样的开发语言开发一个适合我们易库进销存客户的客户端而纠结着。它能用于打印，能识别客户端打印机个数和打印机类型，能查库存，还能用在iPhone、iPad、Android手机上，一次投入，到处运行，我该如何选择？



邢波涛
北京新软孚信息技术有限公司技术负责人。关注SaaS管理软件和B2B、B2C电子商务的融合。

开源的系统级编程利器：D语言

9月的TIOBE语言流行度排行榜中，支持内嵌汇编的D语言，以其系统级编程的特性，被越来越多的开发者选用。

在Digital Mars公司的D语言首页上，放置着这样一段话——“计算机编程语言门类众多，其来源大致可分为两种途径，一类新语言来自于极具创新精神的学术界；另一类则源于关注研发结合和互联网的新技术公司。目前正是在实现编译器的实践经验中诞生一门新语言了。”在九月的TIOBE语言流行度排行榜中，正是这门支持内嵌汇编的D语言，以其系统级编程的特性而被越来越多的开发者选用而重新入围了排行榜前二十位。

D语言是由Digital Mars公司开发的面向对象、指令式、多范型的系统级编程语言。诞生于1999年的D语言，在其设计开发十年后的2009年开放了D编译器DMD的源代码，包括D 1.0和D 2.0的编译器前端和后端源代码，分为Windows、Mac和Linux三个版本。编译器前端以GUN GPL和Artistic开源协议授权。

可以说，D语言是由数十年来实现多种语言编译器的经验积累而形成，源自C++，并为改进和超越C++而生。D语言对C++的部分特性进行了重新设计，其模板在C++的基础上做了扩充，并在语法设计上借鉴了Java、C#等多种流行语言的优点。D语言在提供C++这样强大能力的同时，又具备与动态语言相媲美的开发效率。

C++应用广泛的一个重要原因是C++具有进行操作系统底层编程的能力。D语言同样也提供了底层编程的特性，并加入了完善的内嵌汇编支持。内嵌汇编的特性让系统级开发者能够在标准D语言代码中嵌入目标平台的汇编指令，来完成高级语言难以实现的工作。这些处理任务包括底层驱动程序开发、获取任务堆栈信息、获取SP或RA等寄存器的状态、视频解码等各个层面。除此之外，在某些特定的数值运算中，如带进位加法运算，开发者常嵌入汇编指令来增强程序代码执行的效率。

D语言的系统级编程特性，成为D语言区别于Java、C#等其他高级语言的主要特性。例如，Java语言在语法级别上不提供内嵌汇编的支持，如果需在代码中嵌入汇编指令，只能借助于JNI的Java本地调用，通过引用C/C++代码来间接嵌入汇编指令；而D语言则可以直接与只接受汇编指令的处理器进行交互，直接执行汇编指令集来保证操作的精确性和原子性。

尽管现代的高级语言编译器已在编译优化上做了足够大的功夫，但高级语言编译器为了保证通用性，往往只编译生成通用指令集，没有针对特定处理器的专用指令集进行优化，且无法生成新型处理器增加的高性能操作指令集。无疑，D语言将是内嵌汇编的最佳语言，它带来的便利让开发者撰写最高性能代码来充分发挥硬件潜力变得触手可及。



高昂
中国标准化研究院助理研究员，从事信息技术标准化研究工作。关注开源社区，也是OSGeo中国和InfoQ中文站成员。

手机是怎样变成手雷的

越来越多的人开始关注移动安全问题，矛头直指各种恶意的应用，这说明手机安全问题已经浮出水面。

9月，影星Scarlett Johansson的半裸照开始在网上传播，当事人已经向FBI报案，看起来这只是今年3月传出的好莱坞多名影星的电子邮件账户等遭到入侵事件的延续。但一篇未经确认的分析已经在网上流传，即一部分攻击可能源于攻击者在机场设置了仿冒基站或AP，并以此为手段将被劫持的手机引导到一个仿冒的iOS注入的页面。这算是给本月非常火热的移动安全话题再度加入了一些佐料。

就像当年的桌面反病毒需要把大量非官方预装版的天生可能带有安全漏洞和流氓软件的系统版本作为一个既定事实来看一样，手机安全厂商亦面临着大量水货智能机的非地下罐装版本的侵袭，而更尴尬的是，恶意代码和流氓软件一经刷入，就变成通过一般方式无法卸载的东西，对应的，安全软件反而处于权限不对等的局面当中。看起来为了获取更高的装机量，似乎手机安全软件的发布者也需要和这些地下OS来进行某种交流。这种大陆产业独有的怪象值得思索。

我分析了这样一个有趣的应用——某个“地下电子市场”的实现，这是一个仿冒Android Market的应用。但其原理十分有趣，它自身并非是一个直接的恶意扣费、刷流量的程序，而且其数据获取源确实来自Android Market。改变的就是Android Market的排名，可能把自己提交的或者向其支付某种费用的应用，排在前面，其中大部分是一些中文色情书籍和成人软件，这些软件确实是在Android Market提交成功了的。这样就不存在会被手机提示是否安装非Android Market获取的程序的问题。同时，还套取了用户登录Android Market的账户，之后可以再利用这个账户完成类似给某个软件好评的工作，从而形成一个体系。

这种应用的存在，比目前国内出现多个第三方电子市场，还要危险，因第三方电子市场为了保证自己的信誉，往往还要承担一定的安全审查责任。但这种“地下”电子市场，则完全归于与地下经济的产业链条，只要其装机量足够大，任何一款带有恶意应用或者提权的软件，在通过Android Market并不严格的安全审查后，无论其多么不知名，都可以通过这种地下市场的推荐达到很高的装机量。而过去我们在评价类似Android Market遭遇恶意提交时，并没有考虑到类似事件的影响。Google也看似做出了应对，但其应对是否能到达这种用户呢？从安全的角度来看，这不仅是一条不可控的通道，而且还具有“放大器”的价值。

综合来看，手机安全的尴尬是因其面临过长的不可控的产业链条——从被曝光的某固件厂商问题，到其预装、零售的环节——有大量不可靠不可控的节点，而从传统非智能机市场的大量ICP来看，其驾轻就熟的恶意扣费线路也必然将与智能机合流。而其安全态势的威胁则更多的不是面对传统PC平台早期的病毒、蠕虫、木马等，而是直接和应用进行合流。攻击者扮演着应用篡改者的角色，或者就是扮演着应用开发者的角色。而同时传统PC的安全威胁和手机安全威胁的合流化趋势也十分明显，如基于autorun和sync等机理进行跨平台的交叉感染；和Zeus（PC端）+Zimto（手机端）的组合来讲用户的网银账号、密码与手机确认码采集在一起来完成恶意行为。

当然手机安全虽然是热点，但不是唯一的热点，本月大量曝光的CA问题，让安全工作者有数字签名体系正在崩塌之感，uTorrent软件也一度被替换为假冒反病毒软件的恶意程序。而Linux kernel服务器被入侵，导致者Linus Torvalds进行了源码“搬家”工作。如果整个世界已经是一座燃烧的军火库，那么手机变成手雷也是必然。而本月让我感觉欣慰的一件事情则是，由于利用防盗软件，失主获取到了自己所丢失笔记本的使用者的照片等信息，并进行了张贴。于是开发防盗追踪软件的公司、失主和警方都被购买了这台“2手”笔记本的使用者起诉。法院的判决是——虽然软件开发者可能在帮助校区追回手提电脑方面有着高尚的目的，但“一个明理的陪审团发现他们越过了非法边界。”

安全工作者从不是世界的拯救者，只有在法制精神持续得以伸张、隐私权利持续受到尊重的大前提下，他们才能是勇敢的排雷工兵，而不是炮灰。



肖新光
网名江海客，安天实验室首席技术架构师，研究方向为反病毒和计算机犯罪取证等。微博：weibo.com/seak。



Windows 8 大变局

策划 / 孟迎霞

微软BUILD大会于美国时间9月13日在加利福尼亚阿纳海姆会议中心召开，这是一场面向微软开发者的技术盛宴。全新Windows 8开发人员预览版的发布是本次大会焦点，它对于Windows的发展具有划时代的意义。

BUILD：Windows 8启动开发者新纪元

文 / 蒋涛

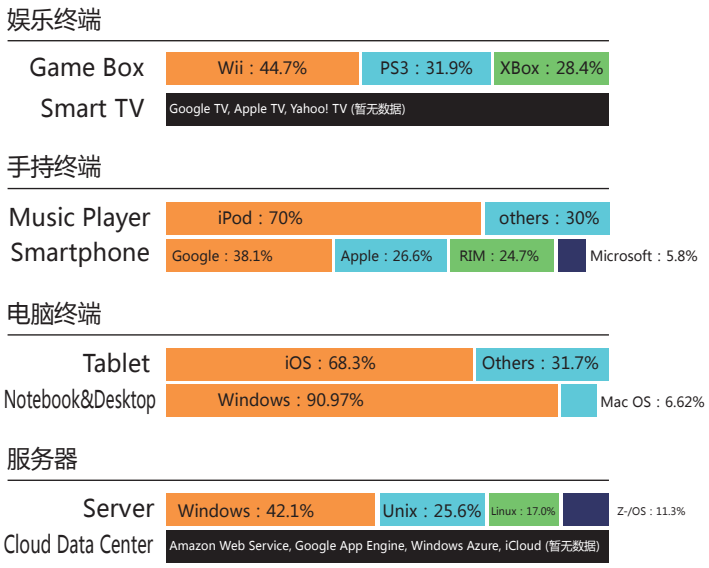
9月13日～16日在加州阿纳海姆会议中心，来自世界各地的超过5000名开发者齐聚一堂，不时为BUILD大会Windows 8新特性的主题演讲发出阵阵掌声。这个传统上微软为其开发者举办的名为Professional Developer Conference（简称PDC）的会议，这次更名为BUILD。

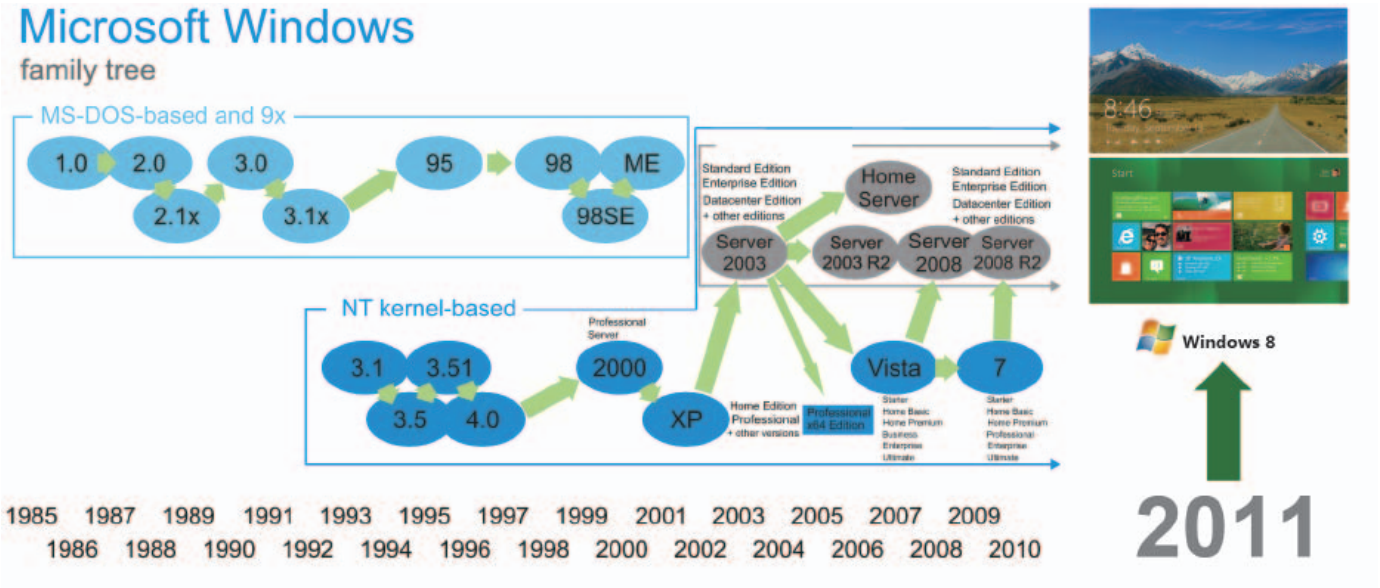
Windows是世界上用户量最多的操作系统，2009年10月推出的Windows 7销量超过4.5亿（加上Windows XP和Vista的用户超过10亿），围绕着Windows聚集了近百万的硬件制造商、设备制造商和软件厂商以及其他独立厂商，这是价值数十万亿美元的IT生态系统。

这个生态系统的核心是开发者，只有各厂商开发者和独立开发者基于Windows API开发出各类应用，Windows才能给用户创造出更大价值。

Windows系统的市场占有率超过90%，唯一竞争对手Mac OS仅有不到6%的份额。但从2007年苹果发布iPhone手机和iOS操作系统以来，iOS的设备用户已超过2亿，iOS的应用数量超过50万，2010年iPad发布后持续热销，2011年新版Mac OS X Lion也引入触摸手势操作TouchPad，IT终端系统在Windows统治三十年后开始发生巨大变革。

我们可以看一下IT终端系统的格局图：





微软的大本营在PC电脑终端上，其营收和利润主要来源于Windows和基于Windows的Office，最近两年的收入增长全部依赖Windows。

iPad持续热销，不仅风靡个人应用，甚至已经开始侵蚀PC企业市场。在2011年Q1财报电话会议中，苹果宣布有超过80%的财富100强企业已经开始采用iPad。有分析家认为iPad在未来两年会侵占20%甚至更多的PC市场份额。

如果微软在PC端的份额也开始下降，那意味着微软帝国的基石将被动摇。面对这一严峻挑战，微软必须给予强力反击。在BUILD会议的开场演讲中，微软高级副总裁、Windows大主管Steven Sinofsky（之前是Office的总经理）披露从2009年夏天就已经开始规划Windows 8（在Windows 7发布之前）。从一开始其目标就是颠覆已有的Windows，从用户体验到核心技术，直到芯片组。

其成果是Windows 8推出了全新支持触摸控制的Metro Style Application。Metro风格的应用使用起来流畅快捷，用户体验已和iOS大大接近。而Windows 8的开放体系结构，将支持大部分的外设，同时Windows 8也支持原有的Windows桌面应用，数十万现有的Windows程序仍将可以运行。这是Windows 95有史以来最大的变化，虽然有人在微博上评论说每次微软发布新版Windows都要说这一句。但这一次是真的。

历史上最大的开发者会议是1993年微软的PDC，地点也在阿纳海姆会议中心，超过8000名开发者参与了Windows Chicago（Windows 95的代号）的亮相。这次微软选择同一场地来发布Windows 8正表明了他们的良苦用心。

其实微软所面临的情况和20世纪90年代初期更为类似。80年代图形界面已成为明显的趋势，苹果在Steve Jobs的主导下推出了Macintosh，IBM在研发OS 2。而微软的大本营是DOS，微软秘密研发Windows，在保持DOS应用兼容的情况下1990年发布Windows 3.1而大获成功，Windows 95的发布更是奠定了微软在操作系统的王者地位。Windows的成功依赖于DOS应用程序的兼容和GUI界面的重大改进。

现在用户体验成为产品发展趋势，Windows 8引入Metro界面系统，同时兼容Windows 7应用程序，正是同一战略的实施。针对开发者，微软保持了开发工具的一致兼容性，原有微软系列开发语言Visual C++、C#、VB都得到了良好的支持，并强化了HTML5的作用。

发布会当天在阿纳海姆乘坐出租车，司机得知我们要去参加微软的Windows 8大会，他拿出iPad说微软要出这个吗？我们说是的，三星出产的Windows 8平板电脑，可以用微软的Office。这个司机很高兴地表示晚上回去要看视频研究下，因为他平日工作主要用Office处理文件，而用iPad并不方便。

这是一个有趣的花絮，除了用户体验，应用

程序的兼容性对用户也是至关重要的。

相比20世纪90年代初，Windows已建立的生态系统更加强大，如果能保持生态系统的迁移升级，Windows系统将重新焕发活力，这也是微软把本次大会重新命名的深刻原因，Reimagine Windows。

终端形态的改变给所有PC硬件厂商和软件厂商都带来了巨大机会，想象下未来三年，大量的新终端设备产生，随身携带的手机，体感识别的游戏机，触摸的平板电脑和笔记本电脑，大屏幕的智能电视，大部分的应用程序都将被重新塑造，新终端的使用还会产生更多的新应用场景，Windows Store和Azure Market Place更是给开发者提供了更广阔的市场空间，

微软CEO鲍尔默在第二天的Keynote终场用这样的话结束：“There has never been a better time to have software development as a core skill.It's the day and age of the developer—it's the day and age of the Windows developer.”

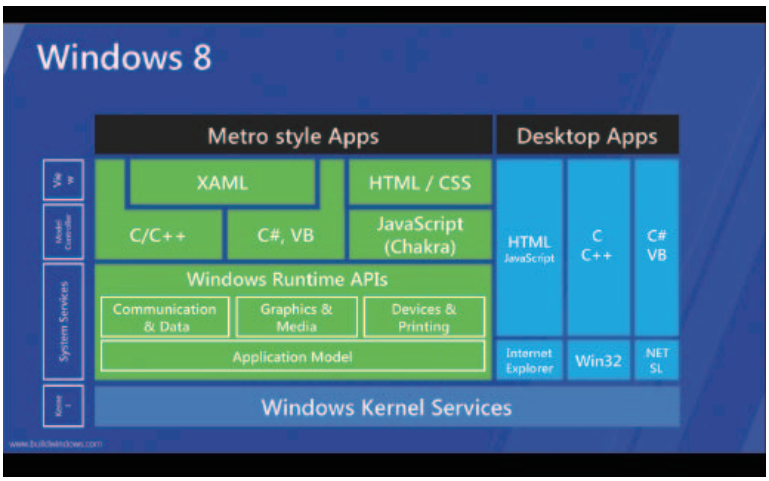
这是属于开发者的黄金时代，对于Windows开发者，这将是新纪元的开启。

以 Metro为名的用户体验

文 / 黄勇坚

本次BUILD大会重点围绕Windows 8打造的新平台、新技术和新理念展开，号召开发者为下一代Windows平台的多点触摸技术和以触摸交互为中心的Metro样式应用程序作好准备。

大会以Windows 8为中心，在首日微软就向开发者展现了新一代Windows 8的多点触摸用户体验和新Metro样式的应用程序。同时我发现Windows 8的开始菜单和Windows Metro样式的应用程序与Windows Phone 7平台的极为类似，新Tile风格的开始菜单允许用户自定义布局，以多点触摸方式进行缩放，同时支持在菜单图标上显示应用程序的信息。桌面App和手机App在用户体验和风格上的类似，体现了桌面和移动终端App有同化的趋势。随着移动终端性能的提升和3G网络的普及，传统互联网和移动互联网的应用在趋同，到了Windows 8年代连App的样式也在趋于相同，PC和移动终端的界限正变得越来越模糊。微软还宣称将要运营Windows Store，为Windows 8的桌面App开发者和最终用户提供App交易服务，这将为Windows 8桌面开发者带来一种新的盈利模式。微软也在即将发布的Visual Studio 11的一级菜单栏上直接添加了名为“Store”的菜单，允许开发者为即将在Windows Store上发布的App进行各种设置。微软在Visual Studio如此显眼和重要的位置加入这个菜单，可见微软对Windows Store的重视程度。苹果App Store的极大成功一直刺激着微软，微软从移动终端的Windows Phone 7就直接复制苹果的App Store模式，而现在在桌面领域也这样做了。



Windows 8开发平台和工具架构图

新的Windows 8平台，为用户带来了全新的多点触摸式体验，为桌面App开发公司提供了新的盈利方式，在技术上，微软需要引入新的API。Windows 8推出了新的Windows Runtime API（简称WinRT API）。与传统的Win32 API相比，新WinRT API在风格上与.NET类库是一致的，WinRT API绝大部分的类都在一个全新的名为

“Windows”的命名空间下，开发人员要是不看命名空间，在风格上是感觉不出来哪些是.NET类，哪些是WinRT类。WinRT API是操作系统的一部分，也是Windows的API。当年用过Win32 API的人，都知道当时使用Windows API是一件很麻烦的事，而到了Windows 8年代，使用Windows API却变得如此轻松，甚至Windows操作系统还能直接识别String、Int32这一类的基础托管类型，这真是开发人员的福音啊。

在Windows 8的前台界面开发技术上，微软推荐使用两种方式：一是使用XAML，二是使用HTML5+JavaScript。在BUILD大会上，可以看到微软对HTML5的支持作了很多努力，展示了使用HTML5和JavaScript开发的界面是如何与使用C#和WinRT API的界面进行沟通的，同时在新的Visual Studio 11上针对HTML5和JavaScript作了众多支持。其实HTML5在一些移动平台上已经很流行了，现在连Windows 8都加入了HTML5的大军，使开发者可以使用HTML5开发标准的Windows Metro样式应用程序，这让我意识到HTML5作为跨平台标准界面技术将会大有作为。Windows 8对HTML5的普遍支持和依赖是本次大会留给我的最深印象。

为了支持Windows 8的开发，微软在开发工具上也进行了革新换代。新的Visual Studio 11使用Internet Explorer 10的JavaScript引擎，支持HTML5标准，在智能提示上比以往更精准、更丰富，多了几个工程模板以支持Metro样式程序的开发。新的Expression Blend 5可以直接编辑HTML5和CSS 3，当台上演示这个功能时，台下立刻激起了阵阵掌声。

在参加BUILD大会短短的四天中，我了解到了下一代Windows 8的多点触摸新体验和Metro样式的新应用程序，以及微软下一代开发平台Visual Studio、.NET Framework和语言方面的新特性，还有HTML5的重要地位。.NET 4.5的诸多新特性和WinRT API已经让我跃跃欲试了。我相信提前了解到这些新技术的开发人员，也会和我一样，想尽快试验和体会这些新技术吧。



黄勇坚

国双科技技术研发负责人，致力于微软技术平台上的研究和开发工作，在.NET平台上提供Web、视频和移动App的用户行为分析产品和搜索引擎关键词广告投放管理和优化产品及服务。



本文作者和Anders Hejlsberg的合影

参会小记

在大会的最后两天，我根据个人兴趣，选择性地参加了一些沙龙。这完全是与微软内部一线架构师和产品开发者进行技术分享和交流的大会。我主要听取了Windows Azure、C#和.NET Framework的发展方向和新特性、Visual Studio 11的新特性以及其为软件开发过程提供开发协作支持的新特性等技术沙龙，当然还少不了去听关于C++发展情况的内容。最令人惊喜的是，期间还遇到几位大师，分别是Windows内核研究圣经“Inside Windows”系列的作者之一Mark Russionovich，这位大师当前正在为Windows Azure的Fabric Control搭建架构；还有C++大师Herb Sutter，曾在大学时我就经常阅读他的著作，这次听了他的讲座之后，才知道这位大师原来还是一位演讲专家；还有就是传说中的Delphi和C#之父Anders Hejlsberg，并有幸和这位平易近人的大师合影。

Windows 8 云体验

文 / 涂曙光

微软BUILD大会中，除了全新Metro风格的Start界面，对Windows普通用户而言变化很大的，还有Windows 8与各种云服务的集成。在Windows 8之前，Windows Live一直都是作为一款独立的软件被用户下载安装到自己的电脑上，而在Windows 8中将直接内置Windows Live。每个用户都可以将自己的Windows登录帐号与自己的Live ID绑定在一起，甚至直接

使用Live ID登录计算机。Windows Live会将用户的个性化设置（比如桌面的布局）同步到Live云服务中，当用户使用其他设备时，这些个性化设置就将被应用到其他设备之上。

Windows 8内置的邮件和日历也将直接与Windows Live云服务相连。当用户使用Windows 8内置的邮件和日历功能时，实际上就是在使用Windows Live云服务。不用担心自己日历中的项目被丢失，或是从其他设备中无法访问或同步，实际上，由于这些数据都会存储到Windows Live云服务中，用户可以从任何设备、任何地点访问到这些信息。Windows 8的联系人功能也尽得Windows Phone 7 People Hub的精髓。Windows 8能自动将用户联系人在Windows Live、Facebook、Twitter等Web服务中的信息汇集起来并集中显示在Windows 8的联系人信息中。用户不再需要访问各个Web网站，就能直接在Windows 8中看到自己的联系人的各种更新信息。

除了微软自己的Windows Live云服务之外，Windows 8也内置了其他第三方所提供的云服务。比如，当用户打开Windows 8的图片浏览功能，除了能浏览存储在本地计算机上的图片之外，用户还可以直接查看存储在自己Windows Live SkyDrive、Flickr和Facebook上的照片。如果需要，用户可以将需要的图片直接上载到Windows Live SkyDrive中，以实现备份或共享。

为了让第三方的云服务能更好地集成到Windows 8中，微软在Windows 8中设计了多种标准的操作，包括搜索、发送消息、共享等。在任何一个应用程序中，用户都能将那个应用程序中看到的内容，发送到这些标准操作中并执行某个操作。比如，当用户浏览网页时，当看到感兴趣的内容时，可以直接复制这些内容，然后呼出搜索操作，Windows 8的搜索操作将搜索本地计算机的内容，并同时调用集成在Windows 8中的能提供搜索操作的第三方云服务，执行搜索并显示搜索结果。这种标准操作模式将为第三方云服务提供广阔的平台。

Windows 8的目标之一，就是让用户在使用Windows的过程中，无缝地连接并使用各种Internet上的云服务。这不禁让人想起Google Chrome OS的目标，就是成为一款纯粹使用云服务的操作系统，可是它只允许用户通过浏览器使用各种云服务。Windows 8无疑也认可云服务的价值和重要性，但Windows 8更相信不应该限制用户只能通过浏览器使用云服务，用户使用云服务的方式应该更灵活、更丰富、更多样化，而Windows 8将做到这一点。



涂曙光
微软平台技术专家，专注于.NET、Web、SharePoint、Office等技术领域。现任职于中国惠普有限公司。

Windows 8 引发硬件大变局

文 / 马宁

2011年的微软BUILD大会注定会成为微软历史上的新一座里程碑。在这次大会上微软宣布了Windows 8的发展战略，随后，微软的股票出现了明显的上涨。本次大会的意义可以和2000年比尔·盖茨宣布.NET战略相比。

Windows 8是第一个支持ARM处理器的Windows操作系统，这也正式宣告了垄断PC领域近30年的Wintel联盟破裂。当然，在未来很久的时间里，x86、x64架构的CPU仍然会占据PC市场的主流，毕竟ARM还没有64位处理器。但随着ARM运算能力的提高，将会在移动设备、低功耗服务器两个领域发起对x86架构的挑战。随着智能手机、平板电脑的普及，Facebook等网站移动设备的访问量已经超过了

Web访问量；而随着云计算的流行，数据中心功耗的问题也成为热门话题，ARM在功耗方面的优势符合了环保理念。

反过来看，随着Windows支持ARM芯片，Windows也有更多可能进入之前的嵌入式设备领域，比如互联网电视、ATM机、POS机等。微软之前的策略是利用Windows CE、Windows Embedded Standard，但兼容性不好、缺少硬件支持的问题让这个市场一直不够大。而Windows 8和SSD的组合让启动速度更快，这样也符合了嵌入式设备的要求。长期阻碍Windows进入嵌入式领域的两大问题之一“启动速度”已得到解决，希望另一个问题“蓝屏”也将得到更好解决。

Metro UI的引入，也增强了Windows在移动设备领域的竞争力。计算机外设决定交互方式，键盘会使用字符系统，鼠标会使用GUI，而多点触摸会使用iOS或Metro UI系统。Metro UI诞生自Encarta 95，经历了Zune HD和Windows Phone的洗练，终于回归到Windows。这也意味着Windows也会增加多点触摸设备，所以除了Windows 8平板电脑外，曾经流行过的Tablet PC和PC触摸屏设备也会迎来一个巨大的发展机遇。当然，我们也不能忽略Kinect这个神器，Xbox游戏机已经准备在新版Game Center中使用Metro UI，据谣传使用Kinect的互联网电视也在开发中了。当然，如何让Metro UI与已有应用共存，也是Windows 8面临的巨大考验。而Metro UI的影响力没有仅仅停留在Windows平台上，Google Android最新网站和一些iOS应用也开始采用类似Metro UI中的Live Tile了。

在开发领域，微软也拿出了压箱底的技术。长期以来很多技术停留在产品组或实验室内部，由于各种因素而没有发布给开发者。Windows Phone中饱受诟病的一点就是不支持C++开发。而这次Windows 8的RT Framework则一口气支持了.NET、C++和JavaScrtipt，之前长期被困扰的Silverlight社区也终于松了一口气，Windows 8仍旧支持Silverlight，不过这项技术将成为应用开发技术，而Web开发会让路给HTML5。值得一提的是，C++版本的RT终于摆脱了COM的阴霾，面向对象的API设计将让C++重新受到应用开发者的宠爱。不过C++的内存访问让安全隐患依旧存在。而JavaScript版本的RT，应该会类似于Palm WebOS的方式，为JavaScript增加一些扩展，使其可以调用系统级功能，不过估计会是一个功能子集，JavaScript也会造成一系列的安全性问题。

总之，Windows 8的进步远远超过Windows 2000以来所有Windows操作系统的变化，但进步也意味着风险，意味着损失了向下兼容性，Vista的失败还历历在目，希望Windows 8不要重蹈覆辙。



马宁
微软最有价值专家，Windows Phone开发者。

Windows 8，“不折不扣”的平台

文 / 王立楠

刚刚在加州阿纳海姆落幕的微软BUILD大会，又一次让我们感到了微软强大的生命力。历来擅长后来居上的微软，在过去几年里屡屡在移动市场失利，一度让人感到其在加速落寞。即便Windows 7和Xbox 360 Kinect等产品创造了惊人的销售业绩，市场仍然无法相信，在后PC时代，微软能凭借Windows继续生存。然而Windows 8在BUILD上的全面曝光，人们终于相信Windows不仅依然能够生存，并能再次改变世界。

对最终用户而言，Windows 8真的可以



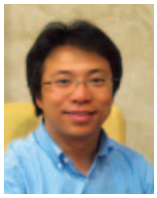
来自世界各地5000多名开发者齐聚微软BUILD大会

称作“不折不扣”的平台。微软正是用“No Compromise（不折不扣）”一词作为Windows 8研发的定位。这是针对苹果公司在桌面操作系统Mac和移动操作系统iOS采用了完全不兼容的设计理念而言的。苹果的思想认为，鱼和熊掌不可兼得，为了保证最好的体验，靠鼠标键盘操作的桌面操作系统和以触摸为主的移动操作系统不应该融合。然而，对于用户而言，平板设备依然难以完全代替桌面电脑，有了平板也扔不掉桌面机，不兼容给用户带来了不便。微软这次是要告诉苹果“鱼和熊掌可以兼得”，Windows 8真正地把桌面和平板完美融合，不折不扣。

Windows 8还是一个“与时俱进”的操作系统。Windows Phone 7手机操作系统在社交网络和云服务上的深入整合，已经给市场留下了深刻的印象，并引发了市场对互联网与操作系统关系的全新思考。苹果公司不久前也宣布了iCloud，使云服务与操作系统的整合更加清晰。然而，苹果还是留了一手，iCloud是苹果自己的服务，且只提供给iOS用户，第三方云服务在苹果的平台上没有机会，只有Twitter幸免，因为苹果没有自己的社交网络。而微软在Windows 8上再一次应用开放战略，第三方社交网络和云存储可以深入整合。其先进的设计构架允许应用开发者使用通用的代码即可无差别地访问各种社交网络和云存储。对于互联网开放平台而言，其API不再是开放给开发者，而是操作系统。而对于最终用户而言，在某一个应用里选择连接哪一个社交网络和云存储不再由应用开发者来决定，而是用户自己。例如，应用程序需要打开云存储中的文件时，开发者只调用一句代码“Open Picker”，用户就会获得一个由操作系统提供的界面，显示所有已经整合到操作系统中的云存储列表。操作系统整合云服务将使社交网络和云存储成为用户真实生活和工作中的一部分。社交网络将作为操作系统通过人与人的沟通来紧密绑定用户的中枢神经系统，相反，操作系统也将促进社交网络向一个方向发展——实名化。你可以在互联网上有一个虚拟的身份，但你还需要一个实名的社交网络帐号，因为你的生活和工作已经离不开互联网，实名社交网络很快将从需要变成必须。像人人网这样的实名制社交网络将是Windows 8开放整合的

首选目标。

上面的一切给了每个用户继续使用Windows 8的一些必要条件，而达到充分条件还需要一个重要的因素——应用程序。谁来最快地给Windows 8开发应用呢？微软这一次给出的答案不仅是C++、VB和C＃开发者，又加上了一个庞大的Web开发者群体，因为Windows 8的原生客户端应用可以使用HTML＋JavaScript开发。这一点对移动应用开发饥渴许久的Web开发者而言，可谓雪中送炭。苹果和谷歌在过去几年中凭借App Store模式为移动应用的开发者开创了庞大的新天地，桌面应用开发者也凭借开发语言和开发平台的近似，大量转行进入移动应用市场开拓。然而，苹果和谷歌在应用运行平台和开发工具方面的积累不够，始终不能为Web开发者提供进入移动应用市场的机会。而微软和Adobe虽然是应用运行平台和开发工具的老手，但苹果和谷歌舍不得让出这块田地。现在你明白为什么乔布斯一个劲儿地说Flash不好，自己支持HTML5却至今也离不开浏览器，做支持Web开发体系的原生客户端应用平台真的没那么容易。但Web开发者的大军无比庞大，曾经在很多平台被炒得大红大紫的Widget框架，就是为了吸引Web开发者做客户端应用。然而Widget毕竟不是原生客户端，在性能和安全限制方面都有局限，Widget始终未成气候。而Web开发者也已饥渴至极。虽然，经常能听到某Web开发团队转向iOS或Android开发，但Web和移动平台的巨大差异，还是让大多数的Web开发者望而却步，从心理上就有难以逾越的障碍。而微软这一次正是知难而进，利用自己即有操作系统，又有应用运行平台的研发能力，投入巨大的资源重写Windows应用的运行时环境，即WinRT，把使用C++的Win32和使用VB、C＃的.NET统一在一起，同时还加上了HTML＋JavaScript，又是一个“不折不扣”。这一次，Web开发者不用担心去学习完全摸不清头脑的Objective-C或使用蹩脚的ADT，只需使用熟悉的语言和熟悉的工具就可以在浏览器外面做应用了。Web开发者，积极行动吧，这次Windows 8的蓝海有你们一份了。



王立楠

2004年加入微软，微软七年的开发布道师背景。从智能客户端、Windows Mobile、Windows 7到Windows Phone 7，经历了微软客户端产品的发展历程。2011年5月加入人人网，任首席布道师。

微博热议 微软BUILD大会

Windows 8全兼容Windows 7；内存消耗大幅降低；全新Metro Style App；可用现有微软的开发语言VC、C#开发；Windows Store即将发布；原Windows软件也可加进去。

——CSDN和《程序员》创始人蒋涛

Metro UI应该说不只是一层皮，最关键的是提供了一套App Framework，新的App可以利用Metro实现PC、Tablet和Phone的三界覆盖，大大降低开发和移植成本，缩短开发周期。老应用可以继续基于Win32，想用Metro就得改写啦。

——微软中国资深架构师沈强

Windows 8正式发布，微软给大家展示了眼花缭乱的新技术，有无数的创新点可以让用户为之欢呼，但整体感觉还是没有脱离Windows系统复杂的界面和操作过程，全融合的概念固然美好，但不区分使用场合所带来的复杂性将让其效果大打折扣。

——广东联通产品创新部副总经理李韩

微软应该通过Windows 8打造全新的生态链：这个生态系统需要有统一ID、盈利丰厚的App Store、丰富无缝的互联网服务、支持各种设备、屏幕、硬件架构……如果此举能够成功，微软满篇皆活。现在看，至少微软是在按这个方向走。但我最担心的是他们的互联网服务还是跟不上。

——CSDN和《程序员》杂志总编刘江

在BUILD大会首日上午，微软介绍了Windows 8的新功能和新用户体验。新版本有一套称为Metro的样式，和Windows Phone界面一致，桌面和移动程序趋于同化，强调触摸体验，桌面程序不经修改或稍经修改即能在移动设备上运行。HTM5和Silverlight是主要表现层技术。

——国双科技技术研发负责人黄勇坚

Windows 8为什么能启动这么快？采用SSD是一方面，更重要的是Windows 8采用了混合启动的方法，也就是仅把很小的系统内核在关机时写入磁盘，这样冷启动时可以从磁盘中快速恢复操作系统内核，然后再加载用户进程，这对绝大多数机器都是有效的（快30%～70%），不管有没有SSD。

——微软亚洲研究院主管研究员张磊

2011年分布式版本控制系统流行度大调查

文 / 青耘

什么是分布式版本控制（DVCS）？它是一种不需要中心服务器管理文件版本的方法，但是它也可以使用中心服务器。其更改可以被合并到DVCS的任何其他用户系统中，因此可以实现非常灵活的工作流。相对于集中式的版本控制系统，它的主要优点是版本控制灵活、支持多种工作流、速度更快、大多数操作可以直接在客户机本地进行。

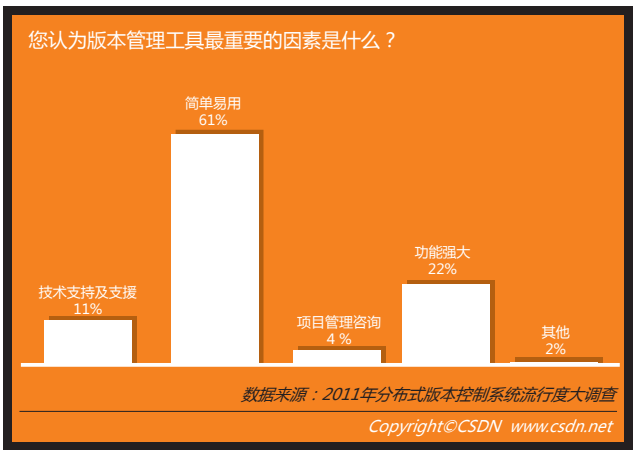
95%开发者用过专业版本控制工具

那么它在中国开发者用户中的应用情况如何呢？为此，CSDN网站做了一个“2011年分布式版本控制系统流行度大调查”，可以一窥DVCS工具的使用情况。在发布网上调查问卷一个多月后，CSDN累计回收问卷近千份，其中来自北京的反馈占26%，其他主要分布在上海、深圳、成都、广州、杭州、西安、大连等城市。这其中，约95%的受访者表示使用过专业的版本控制工具。

在目前正在使用的版本控制工具中，人气最高的是SVN（Subversion），占投票人数的77%。SVN是近年来崛起的版本管理工具，是CVS的接班人。目前，绝大多数开源软件都使用SVN作为代码版本管理软件。SVN正式诞生于2001年，当时设计团队给它设定的目标是在功能上可取代CVS，并在修正最明显的瑕疵的同时，还要保留相同的开发模式。SVN在使用上和CVS很相似，任何习惯CVS的用户都可以很快上手。经过近10年的发展，显然SVN在中国用户心目中积累了不俗的人气。紧随SVN其后的，分别是Git（23%）、VSS（21%）、CVS/CVSNT（19%）。

从具体的使用情况来看，“易用”是大多数用户对DVCS工具的普遍印象，72%的用户认为目前的版本控制工具非常好上手，仅有3%的人觉得它难用。在使用频次上，56%的开发者每天都使用，25%的开发者熟练度已经达到了专家水平，能够给其他使用者一些建议和指导。仅有7%的使用者认为自己需要别人的指导才能顺利完成软件功能上的应用。显然，这样的应用频次和熟练度与版本控制工具的易用性不无关系。

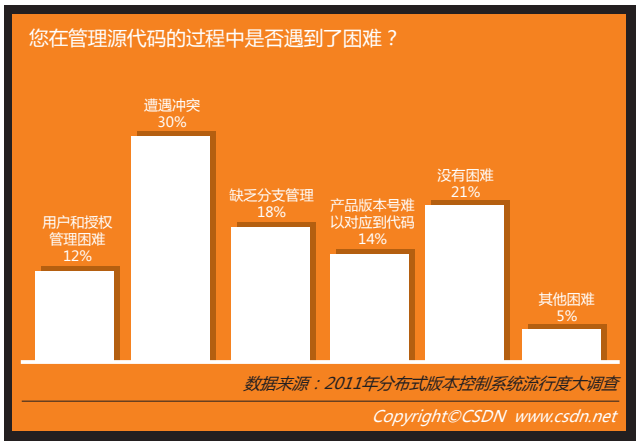
在回答“使用版本控制工具的主要原因”时，62%的用户承认就是因为其“简单易用”，出于其“功能强大”而选择的用户仅占22%。而在正在使用的软件配置管理工具的使用满意度上，感到满意的用户达到了67%。



从使用领域看，绝大多数用于工作上的项目，其次是私人项目、专有项目、Web应用、文档数据管理等。升级软件配置管理工具，是开发者在使用的过程中最常使用的操作之一。调查显示，使用最多的方式是通过二进制文件包进行升级，占46%。紧随其后的两种方式是源代码包（17%）、资源库（11%）。部署版本控制工具的操作系统环境，和大家的直观判断一致，微软的Windows比重最高，占据50%，其次是GNU/Linux，比重是26%。

最大困难是“遭遇冲突”

当然，用户使用版本控制工具并不是一帆风顺。据统



计，用户认为最大的困难是“遭遇冲突”，占据30%，其次是缺乏分支管理（18%），认为是“产品版本没有对应到代码”的占14%，紧随其后的是“用户管理和授权管理困难”（12%）。

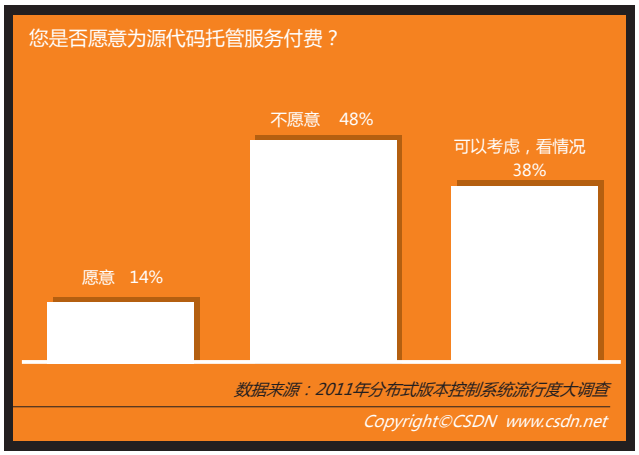
在本地进行管理，是目前大多数用户使用版本控制工具的方式，占据72%，这本身也是分布式版本控制工具的重要优势。除此之外，最重要的方式是第三方托管，占据13.5%。源代码托管是将源代码提供给第三方机构，机构负责保管代码，源代码仅对机构可见，对公众是不可见的。这就好比在银行租了一个保险箱，代码依然是自己的，外界无法得知。从调查数据看，代码托管对于很多开发者来说还是新鲜事物，仅有38%的受访者表示对其有所了解，同时大概28%的开发者有过使用该服务的经历。

对于正在使用源代码托管的用户来说，被使用最多的是Google Code，占据37.3%。根据其官网介绍，通过Google Code，开发者可以：立即创建有关任何主题的项目；获得具有1千兆字节的存储空间的Subversion代码托管以及具有2千兆字节的存储空间的下载托管支持；通过集成的源代码浏览和代码检查工具，轻松地查看代码、检查贡献以及维护高质量代码库；可适应任何开发过程的问题追踪器和项目Wiki通过标星号和更新流，跟踪所关注的项目及开发人员。

位列第二的是Git Hub，占据22.6%。Git是一个分布式的版本控制系统，最初由Linus Torvalds编写，用作Linux内核代码的管理。而GitHub可以托管各种Git库，并提供一个Web界面。根据业界有关专家分析，Git Hub与其他像SourceForge或Google Code的服务不同，GitHub的独特卖点在于从另外一个项目进行分支的简易性，使其为一个项目贡献代码非常简单：首先点击项目站点的“fork”按钮，然后将代码检出并将修改加入到刚才分出的代码库中，最后通过内建的“pull request”机制向项目负责人申请代码合并。

在中国开发者群体里位列第三的是SourceForge，占据22.2%。SourceForge.net，又称SF.net，是开源软件开发者进行开发管理的集中式场所，它由VA Software提供主机，并运行SourceForge软件。大量开源项目，包括维基百科使用的MediaWiki，均在此落户。

那么对于开发者们，选择一项源代码托管服务最看重哪些要素呢？得票最多的是“访问速度”，占据65.5%，看来使用上的便捷性始终是最受青睐的。56.5%的用户最关注的是“是否收费”，54%的用户关注“安全性”，以上三项最受开发者关注，而且其比例数据分布比较平均【注：因为是多选项所以数据也有很大的重合】。看来在使用托管服务时，服务是否物廉价美、稳定可靠，是大家最关注的指标。



过半用户可接受付费代码托管服务

我们再来看看开发者们对于收取托管服务费的态度，愿意付费的用户仅占14%左右，观望态度的约占38%，明确表示不愿意的占据48%左右，也就是说有一半用户能接受付费模式。在大家习惯了免费午餐的互联网时代，这个比例对于那些力图将托管服务商业化的公司来说，相当可喜。而对于具体的费用标准，大多数用户接受的价位在每月90元人民币以内，占据94%；4%的用户能接受90~500元的价位，能接受500~900元价位的则2%。

以上统计，可以初步勾勒出当前的中国开发者对分布式版本控制工具和代码托管服务的心态。总的来说，工具是否易用，服务是否经济实惠，是大家最关心的标准。

勇敢者的变革

记第六届中国敏捷软件开发大会

记者 / 佳琦

自2006年首届大会举办至今，中国敏捷软件开发大会（Agile China Conference）已经走过了六个年头，曾有Martin Fowler、Kent Beck、Dave Thomas、James Grenning、Mary Poppendieck等多位国际级软件工程大师担当演讲嘉宾，听众逾万次，对敏捷方法及其思想在中国的推广和实践产生了深远的影响。

2011年9月1日～3日，由第六届中国敏捷软件开发联盟（CAA）、敏捷方法论的领导厂商ThoughtWorks共同主办的第六届中国敏捷软件开发大会在北京召开。大会以“勇敢者的变革”为主题，探讨如何在企业范围内有效地推广敏捷的最佳实践。

国外大师论敏捷

在第一天的会程里，给很多人留下深刻印象的是一位年逾七旬，却神采奕奕的老太太，Linda Rising，她是《Fearless Change》一书的作者，对于推行组织级大规模敏捷变革有独到的见解。

Linda在50岁时考取了博士学位，对此她说，这是出于一种敏捷的心态，喜欢拥抱挑战，享受从失败中不断学习。与之相对应的，就是固化的思维模式，而敏捷强调的是，我们在过程中可以改进，而不是一成不变。

Linda用几个常见的例子来自证。首先是男孩女孩的不同成长过程，女孩儿从小听话，学习好，家长对此习以为常，认为会一直顺利发展下去。可现实往往出人意料。相反，男孩则从小叛逆，学习差，但在不断地从失败汲取能力后，一个男孩最终可以成长为男人，承担家庭的责任。Linda认为男孩的成长过程是敏捷的。

接下来的例子是经理选拔人才，这里有两种做法：一是把人标识成优秀的和无能的，优秀者总被视为优秀，无能者常常被斥无能，长而以往，优胜劣汰。二是观念中认为，人是不断成长的，从无能到优秀自有其发展的轨迹。Linda认为第二种选拔方法是敏捷的。

第三个例子是团队中有两种人，一种做事顺风顺水，是他人眼中的榜样，但他不喜欢采纳别人的意见，不喜欢团队合作；另一种则做事毛糙，是他人眼中的后进，但却敢于直面自己的不足，不断完善自身。Linda认为第二种人是敏捷的。

因此，虽然在技术活动中，流程、工具、方法，甚至智商很难一蹴而就地得以提高。但是敏捷心态（Agile Mindset），那种不甘于现状、敢于迎接挑战，从失败中吸取教训的心态确是敏捷的基础。

也许在不同文化背景中，Linda的例子不能得到充分支持，但Linda的结论确是经过很多研究和实验才得出来的，如此郑重其事，以科学态度对待敏捷，也表现了大师的风采。

来自英国的敏捷领域大师级专家、咨询师Fred George，以风趣幽默的演讲风格感染了在场的很多听众，很多人笑称他为“老顽童”。他的演讲标题也让人很惊奇：“无政府主义的编程”。在这个标题下，Fred建议在“客户”和“开发者”之间建立起最直接的联系，管理者将为负担与障碍，因为开发者拥有高度的自治能力、很强的工作能力，没有人需要被管理。

现场听众无疑被Fred的做法激动着，因为虽然在国内，出于文化背景、整体技术水平、市场敏感度等情况，这样“激进”的思想不大可能得以施行，但Fred却开启了大家新

的思路，他宣扬道：“恐惧是思考的杀手”，不囿于教条，才会发现敏捷原来还能这样。

国内专家谈敏捷

腾讯副总裁吴军最近新出版了一本讲述IT变革历史的书——《浪潮之巅》，他在其中表现出的学识与洞见令很多人大开眼界。在本次敏捷会议上，他以“金字塔之巅——如何成为最好的工程师”为题，做了演讲。

在《浪潮之巅》里，吴军提到了五级工程师的说法，并给自己的定位是2.5级，但是书中并未详细展开，在敏捷大会上，他给出了完整的答案：

- 五级：能独立完成工程工作；
- 四级：能带领其他人一同完成更大的工作；
- 三级：能独立设计和实现产品；
- 二级：能设计和实现别人不能做出的产品（如Jeff Dean，Donald Knuth等）；
- 一级：开创一个Bussiness（如爱迪生、亨利·福特、亚历山大·贝尔等）。

另外，还有超然其上的零级（如爱因斯坦等人），那是不可望也不可即的。

有了五级工程师的划分，目的在于了解自己所处的层次，也在于了解高手所在的段位，眼界才能宽广。“这山望着那山高”，通往高水平的途径有哪些呢？吴军阐述了其中的三大要素。

时机是成功的第一要素：有记载的两千多年世界历史上最富有的75人中，20%生于1830～1840年的美国，工业革命正在那里兴起，第二高峰是1950年代，比尔·盖茨，史蒂夫·乔布斯属于那个时代，信息革命初露峥嵘。

智商或者基本条件达标是第二要素：就像打篮球需要一定的身高，科学家和发明家要达到智力水平（智商120），但更高以后则不存在明显的优势。

练习达到一万小时是第三个要素：这意味着软件开发者需要五年全时编程经验。

介绍到这里，会发现吴军的演讲似乎和敏捷没有多大关系。然而正如Fred所言，敏捷开发者需拥有高度的自治能力、很强的工作能力，朝向敏捷的改变也是这种基于能力提高的自发行为。无论是咨询师还是布道者，能给予团队敏捷转型的支持与帮助，毕竟有限，所以勇敢者的变革，要从自身做起。

吴军接下来介绍了敏捷实施的三条要则：一、很多事情不需要做，也许某些时候要有壮士断腕的魄力；二、高效的团队小而精，须知往往一个开发小组80%的工作是由20%的人完成的；三、要提高单兵作战能力。



Linda Rising认为敏捷就是喜欢拥抱挑战，享受从失败中不断学习

除了主题演讲以外，来自腾讯搜搜项目的经理总监钱安川，ALU敏捷软件开发讲师、教练何勉，HP敏捷与精益顾问徐毅，上海贝尔企业内部敏捷教练张刚，IBM软件部Rational高级技术顾问许舟平等等，来自实战一线的敏捷专家们，也纷纷开坛论道。在他们主持的七个分论坛中，分别邀请到不同敏捷领域的专家介绍、交流经验。讲师阵容几乎囊括了国内各支敏捷流派，明星专家可谓云集。

总结

敏捷大会举办这些年来，敏捷的概念逐渐深入人心，出现了不少热衷推广敏捷的组织和个人。然而从刚开始培育市场、传播理念，转到近几年的各家争鸣，观点分层，随着组织的敏捷转型和变革越来越热门，大家的困惑也越来越多，甚至出现很大的误区。敏捷在中国是否能真正落地，成为很多人切实的担心。

这其中给旁观者最大的感受是由于各家对敏捷理解不同，很多时候对理念和实践的解释也有差异。在纷繁的“敏捷”中，如何走出一条适合中国的路，也许只有劈荆斩棘的勇士才能做出真正的变革。

软件让创新更智慧

记2011 IBM Rational创新论坛

记者 / 董世晓

在当今社会，软件正在扮演着越来越重要的角色，深入渗透到人类的日常生活之中。世界正变得更加“智慧”，世间万物的感知化、互联化和智能化的程度不断加深。

与此同时，软件已无处不在，深入渗透到人们日常生活的方方面面，导致软件和系统交付过程的复杂程度日益提升。越来越多的企业正在提升自身的软件和系统交付能力，从而在缩短产品和服务上市时间的同时，预测并尽力规避失败的风险。软件正在也将继续使得人类的生活变得更美好。

“软件无处不在”的时代

在8月26日举办的主题为“软件无处不在”的IBM Rational软件创新论坛（Innovate 2011 China）上，IBM软件集团Rational总经理Kristof Kloeckner就谈到了如何利用软件和系统交付平台发挥软件驱动创新的力量。他总结出软件和系统交付转型的三个重要举措：“集成—协作—优化”。集成就是把信息和流程、工具、数据整合在一起，把整个生命周期连接起来，比如将需求管理、测试、认证、开发、部署等，创造整个生命周期的可追溯性；协作就是统一团队、统一思想、统一目标，利用工具提高团队的协作能力，以使得团队可以更快速地做出反应，更加敏捷地反映市场的需求；优化不仅仅是流程的参数，而且意味着商业的结果，它会把项目的执行和业务的目标结合起来。

在IBM Rational首席软件经济学家Walker Royce看来，软件开发更多的是一门经济学意义上的学科，而非工程学科。因为他认为，工程有物理学和材料物理属性的规律，限制了创造力；而在软件中，受限制的则主要是人的想象力。

软件让创新更智慧

随着信息技术的发展，工业产品已经变得更加“智慧”，这使得系统软件的复杂性及开发难度与日俱增。以汽车为例，现在一辆普通汽车都要有数百万行代码，比过去一架航天飞机拥有的代码还多；而一辆功能先进的医疗急救车



IBM软件集团Rational总经理Kristof Kloeckner：软件业的未来依然值得期待

甚至需要超过5000万行软件代码，且在交付使用的15年内还有约1万个软件构件需要更新、测试、部署和维护。所以要保证系统运行，系统软件的开发质量非常重要。

会场上，通用汽车全球研发总监就结合他们在做雪弗兰汽车软件系统的经验，展示了在基于Rational平台上进行软硬件的研发，从设计到上市只用了29个月，要是在过去，这样的项目则需要花费5~10年。

嵌入式软件已经在智能手机、数字电视、汽车电子领域实现重大产业化应用，有分析机构预测到2011年中国市场规模有望达到4650亿元人民币，被国家视作是增强制造业自主创新能力、从“中国制造向中国创造”转型的关键。

IBM软件集团Rational全球系统平台开发及嵌入式系统副总裁Meg Selfe表示：“企业在打造强大的软件开发与交付实力的同时，不仅要着眼于功能创新，同时还应以系统工程的视角，使得软件构件与日趋复杂的电子、电气、机械等子系统智能融合，并且与其他智慧的系统实现高度互联。工欲善其事，必先利其器，软件主宰创新的时代，高效创新的软件开发环境和平台正是实现软件创新的基础。”

开放享未来

淘宝开放平台2011“赢在淘宝”大赛开幕

9月1日，虽已入秋，杭州却骄阳似火，就在当日，杭州淘宝网总部对外宣布2011“赢在淘宝”大赛暨2011“赢在淘宝”应用创新开发大赛正式开幕，该项大赛是淘宝开放平台TOP（Taobao Open Platform）推出的创新选拔赛。大赛期间，开发者通过淘宝开放平台为卖家和买家提供各类应用服务，取得创新成果，参与大赛组织评选，获得大赛提供的丰厚奖励。首先让我们走进淘宝开放平台发展之路。

淘宝开放之路（2009-2011）

2009启程年

2009年6月，淘宝开放平台正式成立，是国内首家电子商务开放平台，依托淘宝用户粘性和明确的业务需求，给公司型开发团队和个人开发者带来了无限的商业机会。从2009年6月22日TOP-beta版发布到2010年4月，注册开放平台的开发者接近万人，上线应用近百个。这一切才刚刚开始，2011年年中，开放平台的统计数字为：上线应用15万，注册开发者达20万。

2010探索年

2010年，开放平台可以说是探索年，从2010年业务方向蜂巢图中可以清晰地看到三个方向：消费者服务（导购工具、生活服务等）、商家IT服务（后台管理，营销工具等）、商家运营服务（店铺托盘、装修拍照等）。像所有的新业务一样，开放平台不是所有方向的探索都一帆风顺，幸运的是开放的脚步并没有一刻停止；伴随淘宝高速成长带来的阵痛和越来越多样化和专业化的诉求下，开放前进的道路越来越明晰，双“十一”大促活动，大部分商家后台系统告急；大卖家热衷于打折促销拉动流量，却不清楚投入产出；一次促销活动，一个旺旺客服同时400个窗口浮动，对买家询问应接不暇；越来越多的营销诉求，官方工具Hold不住，下面，迎来我们淘宝重要一年——2011。

2011淘宝开放年

今天任何一家大品牌进驻淘宝，除了关注淘宝流量，还会关注支撑它在淘宝做电商运营的配套服务，而我们开放的目的正是要完善电子商务产业链，从上到下，从前到后，每个环节都有专人进来做精做深，这就有别于所有以盈利为导向的当今大部分开放平台。围绕淘宝开放平台，电子商务产业链条的新贵也在逐渐崛起：2011年一年，产生百万盈利能力的第三方软件提供商（ISV）50余个，千万级ISV超过10余个。

2011年，除卖家工具市场外，还有两个市场炙手可热——买家和社区电子商务市场。如何满足买家日益专业化和多样化的导购需求，是今年买家业务开放的一个重头戏，也是开放平台从平台底层要支持的重要方向。微博的崛起，美丽说、蘑菇街等站外宝贝分享的垂直社区的炙手可热，促使淘宝开放平台与美丽说、蘑菇街等社区的探索性合作。这些合作在于探索社区电子商务和淘宝SNS化的新方向，并为淘宝的基础“水电煤”服务预热。除了中国境内，值得一提的是一家俄罗斯网站，仅仅依靠开放平台提供的商品、交易、搜索等接口，建起一个境外俄罗斯版的淘宝商城，日均订单1000~2000单；更让人高兴的是，这种模式在中国周边的国家和地区正在不断复制自己。

淘宝开放平台同时要肩负起的重托，是在欣欣向荣的市场下安全和稳定的保障。面对13亿次/天API调用，如何做到精细化的监控和安全保障，是一个世界级的难题。开放平台会积极推动基础服务开放，包括统一应用托管和数据中心服务、性能/安全测试中心服务、监控预警服务，为所有第三方合作伙伴提供安全与数据透明的保障！

淘宝开放平台经过三年的快速成长，已经为百万卖家提供工具和服务！但还无法满足用户多元化的需求，期望通过“赢在淘宝”比赛挖掘更多优质的ISV，并提供强大的产品支持、技术支持，还有丰厚的奖金，来看看“赢在淘宝”比赛的一些介绍吧。

走进“赢在淘宝”大赛

大赛简介

2011年赢在淘宝是以鼓励创新为基调的应用创新开发大赛。在基于淘宝开放平台（Taobao Open Platform）开放业务上，开发者可以开发创新的应用。大赛分为卖家服务和买家服务两个方向，开发者在淘宝开放平台创建相应的应用进行开发。淘宝开放平台支持开发者为卖家提供各方面的应用服务，从旺铺管理、数据分析到后台工具、后台系统。在买家服务方面，开发者从完善购物体验、解决各类导购问题等各种角度创建买家应用，开发者也可以探索更多创新的模式。淘宝将通过评选选出优秀的应用以及合作伙伴，给予丰厚的奖励。

奖项设置

2011赢在淘宝应用创新开发大赛为开发者提供了丰厚的奖励。除现金奖励外，淘宝开放平台联合多家知名VC一起对优质项目展开投资评估，为大赛优质项目提供资金支持，优秀项目还将获得淘宝开放平台孵化计划项目资格。

本次大赛分为卖家服务和买家服务两个方向，各设置前三名、10强以及30强奖励。除对开发上线的应用进行评奖外，同时还将从优秀设计方案中评出创意大奖。

前三名：50万、30万、20万现金奖励

十强：参加商业计划演练、投资评估

三十强：参与优秀项目孵化计划

参赛方式

开发者可以直接登录淘宝开放平台创建应用，勾选参赛标签，即可参赛。开发者完成应用提交审核通过，即成为正式的参赛作品。同时暂时无法实现的应用以设计方案形式参与创新设计奖项评选，此项将于10月中旬开放。

*参赛须知：参赛选手需具备淘宝账号，并经过支付宝实名认证。上线应用均需符合《应用接入审核规则》方可上线，并遵循淘宝开放平台现有各项运营规范。

日程安排

第一阶段：9月1日~11月30日应用开发阶段。

第二阶段：12月1日~12月31日应用评选阶段。

第三阶段：12月底 商业演练及颁奖。

评审办法

大赛将在10月中旬开启作品展示和用户试用、评价等功能。根据用户使用情况、业务创新度、市场容量预期、功能完整和易用性等多个维度，由特邀的资深卖家、买家，淘宝网业务小二、互联网行业专家等评委评选大赛30强。大赛30强均有机会参与最终的商业演练，在此期间需准备团队介绍、商业计划、运营计划等内容，大赛评委将评选出最终十强。前三名的产生：大赛十强可以参与12月底的商业演练及投资评估，决出三强。

后记

纵观国内外互联网的发展趋势，以Facebook为首的开放平台正在引领互联网的潮流，并培育出了市值超过150亿美元的游戏公司Zynga。淘宝网将2011年定为“开放年”，必将推动淘宝开放平台跨越式发展，而“赢在淘宝”大赛更是淘宝开放平台快速发展的催化剂，也必将培育一大批有潜力的公司。比赛的优胜公司并可以得到淘宝网的大力扶持，甚至得到VC的投资，此次大赛将是中国的互联网开放平台的一座里程碑！感兴趣的开发者可以登录http://2011.taobao.com/参赛！





移动·大航海时代

策划 / 本刊编辑部

15世纪到17世纪，以哥伦布、达伽马、麦哲伦、阿美利哥、库克、白令、巴伦支、塔斯曼为代表的欧洲各国航海家前赴后继，不惧艰险，扬帆远航，将人类居住的各大陆连接起来，极大地改变了世界贸易、文化乃至每个人日常生活的面貌，其中多人也将自己的名字永远留在了新土地。这段时期，史称大航海时代。这是全球历史上最重大的事件之一。事实上，我们现在每天都习以为常甚至一些人已经离不开的诸多日常事物，包括辣椒、烟草、番茄、玉米、土豆、红薯、巧克力（取自可可）、向日葵、南瓜等等，都是大航海时代之后才从美洲逐渐进入世界的千家万户。大航海时代更使欧洲快速发展，超越亚洲，成为世界中心和霸权力量。而给非洲和美洲等地居民带来的，则是不堪回首的血泪一页。

今天，随着移动设备的崛起，互联网更加深入地渗透到社会生活的方方面面，Windows+Intel主导的PC帝国解体，世界又迎来了信息产业的大航海时代。从天空俯视而下，港口和近海一带已经拥挤不堪，旧大陆的霸主，崛起不久的新贵，在之前的竞争中失势的小商人，还有更多雄心勃勃但两手空空的淘金者，都得到了远方存在未知新大陆的消息，纷纷来到这里。阳光照耀中，无数的船帆如鹅毛般洒落在海面，有的是豪华舰队，有的只有一块舢板。每个人都被财富、成功的前景和改变世界的热血所驱使，似乎忘记了远洋航行意味着冒险：无人穿越的海洋里没有既定航线，储备的食物根本不够长期航行，木制船壳无法抵抗船蛆的侵蚀和鲨鱼的冲击，同行的冒险者会对自己施发冷炮，甚至早就设下了意在歼灭的漩涡，远方等待着他们的，很可能不是海阔天空，而是惊涛骇浪……

与此前的PC、互联网两次浪潮不同，以智能手机为代表的多样化的计算终端，更自然的用户界面（语音、体感、脑电波……），加上互联网云计算平台强大计算能力和海量数据、智能分析的支持，史无前例地第一次有可能实现无时无刻无处不在地为所有人提供持续的智能信息服务，将极大地改变计算机软硬件、通信、互联网、电子、家电、广播影视、新闻出版、零售、咨询服务等诸多行业的格局。

而对于移动应用开发者们，平台层面的竞争意味着巨大的机遇。我们相信会有新一代的英雄横空出世，在新大陆雄起。本期特别策划，我们采访了多家成功的移动应用团队，分享他们的成败经验，希望有助于更多后来者多避开暗礁、冲破迷雾，行驶在正确的航线上。

百纳信息CTO刘铁锋：

海豚浏览器的创业故事

文 / 楚田风

海豚浏览器不仅在Android Market上大放异彩，被CNET评选为2011年十大最受用户欢迎Android应用之一，而且最近发布的iPhone和iPad版本，也很快冲到多个国家分类排名第一。本文讲述了其背后团队的创业故事。

在移动互联网风起云涌的时代，不少创业者的思路还停留在复制国外成功模式时，百纳（MotoTap）这家成立不到两年的公司，已经在国际舞台上崭露头角。其开发的海豚浏览器不仅在Android Market和App Store上大放异彩，更是被CNET评选为2011年全球排名第二的最受用户欢迎的Android应用。刚发布的iPhone版海豚浏览器，也在不到两天内就冲到了二十多个国家分类排行榜第一名。而iPad版海豚浏览器，不仅仅在近四十个国家分类排行位列第一，更是勇夺iPad美国免费应用排行榜第七名。由于海豚浏览器之前一直面向海外市场，7月才转战国内，很多人并不了解这是一支怎样的团队。

《程序员》邀请到百纳信息公司CTO刘铁锋分享产品背后的创业故事以及其对移动浏览器未来的展望。



海豚浏览器创始人杨永智认为，移动Web Apps即将兴起

做就做大一点！

创业是属于内心坚韧者的游戏。这个游戏需要的不仅仅是追随内心声音的勇气，更多的是面对机遇和诱惑选择的智慧，以及面对不确定性未来的坚定信念。而这个游戏最困难的，是开始的纵身一跃。不是所有的人都有勇气和信心甩开一切背景和前途，投入前途未卜的创业。尤其是对在跨国外企稳定发展的技术人员。

百纳的几位主要创始人均来自微软。工作多年之后，逐渐熟悉了大公司的游戏规则，感受了美式企业的闲适与沉闷，领略了面向全球市场的战略规划以及产品研发要求，适应和把控了跨国公司的节奏与质量要求。他们开始思考下一个阶段的个人发展，参照在微软五年、十年甚至更长年头的资深员工的状态，一致认为这种可以被预测的升级模式，并不是自己的理想工作状态。就如同一部精彩的悬疑小说，在充分理解了其中的所有的疑点之后，就失去了没有阅读下去

的兴趣。

当时正值移动互联网浪潮刚刚兴起，Apple、Google等公司的新一代智能手机平台发布，大力推动了移动互联网的发展。有人甚至高呼2009年是中国移动互联网的元年。

遗憾于已错过的互联网黄金十年，而更具发展前景的移动互联网创业时代正在眼前。几位创始人正视了种种不安分的内心躁动，反复权衡之后，决定利用多年的经验、眼界和能力，在移动互联网领域闯一闯。

创始之初，由于市场思路清晰、产品质量过硬，开发的很多小应用很快就出现在Android Market各个分类排行榜的前十名。在解决温饱问题之后，大家开始反思：是选择赚点小钱，还是进一步挑战自我，真正干件大事？

这时，他们考虑最多的是产业趋势。因为只有认清和把握好趋势，才可能将事业做大。怎样才能够借助移动大趋势充分发展，在风口浪尖跑得最快？什么样的应用既具备一定的技术门槛，又拥有足够广泛的用户基础，而且现有的同类产品离用户期待有较大不足，最好还能不会轻易被竞争对手通过抄袭或者流氓手段所击垮？反复考量之后，浏览器成为了大家一致的选择。看好这个方向的更重要原因在于，浏览器所能够承载的Web Apps会如PC互联网所发生的C/S往B/S迁移的历史一样，成为未来的趋势。

于是，海豚浏览器就这么诞生了。

用心得到的回报

定下目标之后，剩下的就是执行和不断地改进。在2010年初，Android的大量用户都在海外，对于百纳人来说，这并不存在什么障碍。以前就有和老外一起合作的经验，对于把海外的用户照顾好，充分理解并满足他们的文化习俗和消费习惯，一切驾轻就熟。

品质是产品最关键的一环。海豚浏览器的产品设计团队、开发团队在创新上不遗余力。最早为Android用户提供多点触摸、手势操作、提供插件支持、拓展划屏设计等特性，加上



9月份发布的海豚浏览器iPad版

比系统浏览器更快的启动速度等一系列前沿动作，不仅仅让海豚浏览器在海外站稳了脚跟，还击败了不少同一时期的海外竞争对手，海豚浏览器的很多元素和功能也被其它同类产品快速模仿。

除了对产品品质的打造，处理来自于用户的意见和反馈也是团队工作的重要部分。App Store本身就提供了一种非常直接和有效的方式来收集用户反馈。为了及时帮助这些用户解决好反馈以及使用好产品，百纳公司内部默认使用英文邮件交流，海豚浏览器运维人员的入门级要求为英语达到专业八级，同时还需要具备直接与用户口语对话的能力。对于用户的反馈，除了Email的服务方式，还会用IM来联系用户，甚至直接给用户打电话来帮忙他们解决所面临的问题。在用英文一对一给老外解决问题之后，运维人员也收获到超强的成就感。以至于，当运维人员看到“我碰到了xx问题，只能给你们三分，如果你们解决了，我马上评五分”这样的评论，大家会抢着去联系，非常快速地帮用户解决问题。

在百纳内部，对于Market上的评分，不是关注是五分还是四分，而是关注在小数点四位（通过特殊的方式可以获得）以后的变动。每一次版本的发布，是否能够在小数点之后能够带来提升，是否能够一对一地解决所有低评分

用户的所有问题来提升分数。因此，海豚浏览器是为数不多地在Android Market上用户量接近千万，但评分仍然保持在4.5分以上的程序。正是因为对用户反馈的高度重视以及内部近乎苛刻的要求，赢得了媒体的赞誉。国外的专业媒体如ZNet、PC Magazine、Wallstreet Journey等媒体，自发地为每一次海豚的新版本做相关报道。

除此之外，为了做好产品，百纳还直接深入海外来了解用户以及分析趋势，从2010年开始，不管是在美国的CES、Google I/O、WWDC、Facebook F8，还是在西班牙的MWC大会上，不仅有百纳的开发人员，更设置了海豚浏览器的展台。与此同时，海豚浏览器也开始在硅谷开设了办公室，开始直接招聘美国的本地人为我们工作，其中不乏斯坦福博士出身、在美国麦肯锡以及三星工作过的优秀人才，使百纳可以更好地理解和开拓海外市场。

这一切的成绩都在短短一年时间内快速建立。在百纳人的工作准则中，没有给自己设置任何限制，无论国界还是语言的掩盖，都无法阻止他们把产品做到极致。

浏览器的未来

前段时间，百纳获得千万美元级投资的消息受到业界广泛关注，海豚浏览器具有如此高的估值最主要的原因在于创投双方一直看好的移动浏览器未来。

移动互联网的趋势是什么？是多达几十万种各式各样的好玩的小程序吗？是看起来很美好的LBS，是看起来和互联网没有什么区别的SNS，还是流行起来更快的移动支付？这些一直都是业界在思考和探索的问题。

做为一个以技术立足的公司，海豚浏览器更多地从技术角度对大势进行分析。

在PC时代，有不少公司红极一时。但到了现在，除了Microsoft、Oracle、SAP、Adobe之外，获得巨大成功的软件公司屈指可数。而Google、Facebook、Twitter等层出不穷的互联网公司却后来居上，站在时代浪潮之巅。

从技术角度来看，传统桌面软件所代表C/S的开发方式已经逐渐被用户和企业摒弃，缩减到小众的专业领域。而面向互联网服务开发B/S模式则开始大行其道。自然而然的，所运用的语言、开发框架、开发思路，甚至是市场上开发人员的需求，全部转向互联网领域。这是产业的趋势所决定的。

与此同时，得益于Google力推的Ajax技术，一个新的职位——前端工程师开始兴起。Ajax的技术所解决的是浏览器上所出现的刷新问题，让用户不会觉得出现频繁的页面闪动，本质上是用户体验的问题。如果用一个技术概念来映射，这就类似于C/S模式中常用的多线程来提升的用户体验。Ajax带来的，不仅仅是更加丰富的页面交互，更多的在于利用Web技术可能提供类似于Native App的交互体验。这是Web技术上的改进。

此外，从2010年开始，沉寂了10多年的HTML标准开始躁动起来，HTML5似乎一夜之间成为开发者关注的焦点。HTML5其实本质上没有太多革命性的技术，这不过是各家巨头看准了在互联网上新的革命和洗牌机会。仔细想想HTML5为用户带来了什么？新的标签，也就是强制的展示方式标准；Drag & Drop，也就是增强的类似Native App的交互方式；加入新的Canvas的绘图方式，能够直接访问GPU，这就是Native App使用所有的访问硬件资源的能力，再加入剩下的技术特性加起来，不就是之前流行的RIA吗？不过是革了Flash和Sliverlight的命，让原先被这些公司垄断的标准被打破，产生新的平衡而已。这是业界在推动Web技术发生变革。

这些技术的改变，意味着产业趋势、技术趋势，甚至是技术标准已经为Web Apps铺平了道路，也就是说在移动领域一样会出现像PC时代所出现的C/S往B/S转型的必然趋势。

那么，在移动互联网时代，移动+互联网的结合点，就在于Web Apps，承载它的就是浏览器。配合未来上百亿台的移动设备，浏览器会带来更大的想象空间。

这就是专为智能手机打造、结合下一代Web Apps趋势、面向国际舞台的海豚浏览器的未来。

Haypi创始人任刚：

做好推广的核心是产品质量

口述 / 任刚　整理 / 董世晓

在任刚看来，应用推广的核心是产品质量，没有好的产品质量，再完美的推荐最终结果都得不偿失。

Haypi的创业故事

Haypi的创业之路，可以说是既简单又充满戏剧性。

记得那是2000年，我在《程序员》上读到了时任主编的李学凌撰写的那篇名为《到美国去，挣美元!》的文章，文中主角周奕通过做共享软件实现月销售收入超过5万美元的成绩对我触动很大，让我萌生了做共享软件的想法。于是，我们迫不及待地投入了浩浩荡荡的共享软件大军，开始了创业的征程。这一做，就做到了现在。

2007年，我们做出了第一款iPhone共享软件，之所以采用共享的思路做iOS应用，是由于我们之前已经做了很多iPod软件，技术和经验的积累比较丰富，因此iPhone一推出，我们就想到了在它上面做应用。那个软件实际上是个录像机，推出后表现一直不错。但后来iPhone自带了录像机软件，于是我们的那款产品就没落了。

2008年8月，Haypi正式成立。当时我们有两种选择，做一款iOS上的联众游戏或者战争策略类的联网游戏。经过分析，我们认为，iOS本身是一个平台，联众则是平台中的平台，也就是要在大平台上再做一个小平台，将来的发展空间会比较有限，再加之类似平台上的很多游戏与赌博这种灰色产业联系比较紧密，于是，我们最终放弃做iOS联众的想法，选择了做战争策略类的联网游戏。现在看来，这个选择是正确的。我们立志要做第一款能在被窝里躺着打的联网游戏，这种感觉也一直激励着我们。结果未能如愿，当我们推出这款游戏的时候，很多类似的小游戏已经上线了。在做这款游戏的过程中，我们经历了很多事情，修改了多个版本，在上线时又被人说后台结构根本不对，最多只能支持200人同时在线，要知道，我们的目标是支持5000人同时在线。一直以来，我们这个团队最自信的就是技术，如今技术都遭到了质疑，可想而知是比较苦恼的，不过，好在我们挺过来了。

2010年1月18日，Haypi Kingdom正式上线，大批玩家推着我们往前走，这时的感觉就完全不同了。图1是Haypi Kingdom上线以后接近两年时间在美国App Store的排名，当时很多功能没有加进来，也不敢推广。到了6月，后台基本稳定，到现在为止支持4000~5000人同时在线一点儿问题都没有，我们的技术现在看来还是经得住考验的。于此同时，有两个是Haypi Kingdom玩家的投资商找过来，表示对Haypi Kingdom很感兴趣，并进行了投资，于是我们在6月、8月、10月连续做了几次大的推广。

从图1可以看出，2011年的前4个月，Haypi Kingdom排名虽然有所下滑，但一直维持在TOP10。下滑的原因是我们连续4个月没有花一分钱进行推广，现在看来在不花钱的情况下，也有可能维持在TOP 10。



图1　Haypi Kingdom近两年在美国App Store的排名



图2 围绕“产品质量”的众多推广类型

推广的类型

虽说“酒香不怕巷子深”，但要想真正出类拔萃，获得用户的认可和可观的收益，还是得走推广这条路。

我始终认为，推广的核心是产品质量，如果没有好的产品质量，再完美的推荐最终结果都是得不偿失。即使再小的产品也要从质量抓起，如果没有这个“1”，后面无论有多少个“0”也是白搭。图2是围绕“产品质量”的众多推广类型。

Apple推荐，打入排行榜：能够得到Apple的推荐，是小团队梦寐以求的事情，但你只能是被动地等着被推荐，而无法主动地寻求Apple推荐，可以说，这不是一种理想的方式。

限时免费：限时免费曾经非常有效，现在看来效果越来越差，因为每天限时免费的产品特别多，选择太多，导致被用户选中的可能性变小。

FreeAppADay：这类方式的效果非常不错，但资源有限，收费越来越高，而且很难订到合适的档期。

PPC广告投放：用户质量高是其最大特点。

PPI广告投放Tapjoy：这个渠道现在的效果越来越差。

代理发行：主要是Chilingo、EA，这种方法推荐小团队使用，但前提是有很好的产品质量。我们现在不在乎把一部分东西分出去，因为我们现在处于打基础的阶段，通过与代理的合作，能带来很高的用户量。

社交网络：之前我们花了很大精力把它做好，但事实发现这个效果并不是很理想，至少在我们游戏里面不是很好，所以不推荐一开始就加到社交网络里。

传统的互联网广告：不是太理想。

公关：这就像中彩票一样，如果你的新闻稿给很牛的编辑发现了，也对你的题材感兴趣的话，带来的效益会比较大。

交叉推广：目前来看有一些老的推广方式非常不合适了，资金一直在变化，市场在变化，推广方式也一直在变化，所以迫切需要找到一些好的尝试，我认为交叉推广就是非常好的尝试。交叉推广就是把相关的软件放在一起相互推，在2003~2004年时这种方式带来很大的增长。在iOS平台上、在移动互联网平台上，这种交叉推广更好，因为这个平台更统一、范围更小，这样交叉推广大家同时发力。比如说我们有一个大的气泵，平时每个人都往里面充气，在推广时，我要推广给谁就把气筒推给谁，这样的效果非常好。Haypi和Trintit等公司联合推出互助推广联盟，平等互利，非商业化运作，人人为我，我为人人。这个月我们做了三次推广，就用这个平台，都推到了USA Topfree 20之内。

联合运营：就像人人网这样的联运，其实现在已经触及到了移动互联网，我们在与DeNA做联运，让它在本地市场做推广，欧美市场不做联运，其他市场可以交给其他公司做联运。

当然，还有很多其他的推广方式，但目前主要的就是这些。

Haypi Kingdom推广案例

2010年10月的那次推广，我们使用了多种推广渠道，下面就以它为例，为大家分享我们的经验。

推广实录

第一，限时免费：Haypi Kingdom限时收费了一个礼拜，然后突然免费，当时效果非常好，

限时免费将Haypi Kingdom直接推到TOP 100。

第二，FAAD：收费比较高，大约是1.5万美元。

第三，Tapjoy：很有效果，当时花了5万美元，因为是早期的合作伙伴是按0.25美元安装的。

第四，Admob：2~2.5美元安装，Admob算起来不划算，但它的用户质量高。

第五，iAD的前身：在那儿花了1万美元，他们其实也是PPC。

第六，软件内推广：花3万美元在聊天软件TextPlus上，我觉得这个不太划算，当时也没有很好的统计，也不知道3万美元到底带来了多少个安装。

第七，网站广告：我们在35个网站做广告，比如说TA、EP这样的网站，在比较醒目的位置做动画广告条。这个效果看起来夸张的不好，接下来不会考虑在这上面做广告了。

第八，市场公关：我们大约花了1000美元用来做公关。

第九，配合游戏内活动：根据游戏的特性，我们做了一个游戏类的活动推广，有3~4台服务器，这些服务器最牛的买家会在枪战竞技场里面做半个月的PK。这次推广之前我们在美国的TOP 500左右，在推广之后当天效果就比较明显，虽然比较费劲，但能取得这个成绩我们是非常满意的。

推广的最终效果如图3所示。这次推广共花费16万美元左右，带来80万次以上的安装，平均每次安装0.2美元。后来经过统计发现，每个注册玩家平均贡献是1.5美元以上，所以按这个来算，0.2美元换回1.5美元是非常划算的。越往后推广费用会越来越高，市场竞争越来越激烈，推广成本会加速递增。

经验总结

- 重点是协调多渠道同时发力，冲榜成功。
- 每个产品有自己的特点，要依靠数据而非想象来做推广，多做实验，一步步来。
- 推得起，还要立得住。不要冲榜成功就马上撤下所有推广方式。
- 创业初期产品推广

做个好产品：让自己能赚钱，因为不赚钱光忽悠，投资来的钱最终还不是成功的保证。

推广从小的目标国家做起：推广因为花钱比较多，所以一开始先选一个小的目标市场，比如说澳大利亚、新西兰、日本这样的市场，甚至可以是中国（中国市场看起来越来越大了），进行推广。我们一开始就从澳大利亚做起的，到现在澳大利亚还是我们排名最高的国家，你在哪儿用功多市场是会有回报的。

找个好的发行商：这个发行商可以找CocoaChina，它会帮助你们做得非常好，但一切都基于好的产品，如果没有好的产品的话什么都是白搭。

- 适度推广。这个以前提的比较少，我要重点解释一下。既然一次能推到20万下载，那为什么不接着推呢？一个月能达到这样一次效果的话，收入和产出比就会非常好。而那些效果差的推广方式，只会带来高成本，很难带来好的客户。因此，要适度地把推广的节奏拉开一点。



图3　2010年10月推广的最终效果

结束语

Haypi Kingdom上线已经快两年了，我们做的产品的风格、题材差异非常大，每个题材只做一款。这个市场已经到了中后期，已经不是蓝海阶段了，国内作者收入占全球份额的比例已经下降到了6%，要知道，之前的数据可是15%。如果我们继续单打独斗的话，日子会越来越难过。因此我们要携起手来，共同期待中国Happy的明天。

触控科技创始人陈昊芝：

《捕鱼达人》成功的另一面

演讲者 / 陈昊芝　整理 / 陈秋歌

《捕鱼达人》是触控科技旗下PunchBox团队开发的一款手机版深海捕鱼游戏。登陆App Store的首日就迅速攀升至付费应用榜的第一名。大量的技术积累、有效的市场推广是其成功的另类要素。

《捕鱼达人》之所以会取得如此好的成绩，除幸运之外，背后还有一些令人深思的产品故事。

花大量时间做技术积累

在移动开发迅速发展的黄金时代，很难想象一个技术开发团队在近一年时间内不曾推出一款产品，而仅仅专注于底层技术钻研与知识积累。PunchBox却是一个这样的研发团队。从2010年~2011年初，PunchBox团队用了一年多的时间来做引擎、编译器等各种底层的研发和技术积累。



绚丽的《捕鱼达人》游戏界面

直到2011年4月，PunchBox才推出首款iPad版《捕鱼达人》游戏。但推出不久，市场上就出现了20多款仿制类游戏。至今《捕鱼达人》仍占据绝大部分市场份额，用户日活跃量达120万，每日各平台新增下载量有8万多次。这在很大程度上得益于该团队背后深厚的技术积累。其实《捕鱼达人》在最初的游戏设定上就做了与众不同的安排。其中之一就是每小时为用户赠送20个金币。用户在头一天晚上消耗掉了所有金币，第二天一大早就会积攒很多金币；上班时间消耗掉所有金币，下班后又会获得足够多的金币，如此循环，深深吸引用户一直继续玩耍下去。除此之外，该团队还保持每三周，甚至更短的时间内对产品进行升级。

大力度进行市场推广

除了技术上的深度耕耘外，我们在《捕鱼达人》的市场推广上也投入了巨大的财力。在国内，产品一上线，我们就选择走媒体覆盖路线，现在几乎覆盖了国内所有媒体渠道。在国外，比如北美，我们用Google做推广，在近两个半月的时间内，投入近50万美元。这对于小公司来说是无法想象的，而我们在这方面做了大胆尝试。

除此之外，我们也加大了营销力度。我们把“是否为娱乐圈和时尚圈所接受”作为产品是否流行的标准。那流量达到多大，才能引起

时尚圈和娱乐圈的注意，从而流行起来呢？一个半月内500万下载总量，是我们确立的营销目标。时间证明我们做到了，2011年7月1日《捕鱼达人》达到这一目标后，不久就在明星中迅速流行起来。

回首成功历程，我们最大的收获就是，在大胆尝试其他企业不敢涉足的推广方式和推广力度的同时凭借团队一年多来的技术沉淀，我们在《捕鱼达人》一经推出并做推广时，可以迅速保证其流行起来。

抢占空白平台市场先机

一款产品的成功，创新在很大程度上起着决定性作用。通过我对2011年全球TOP 50手机游戏开发商排行榜的研究，我发现他们之间都存在三个共同点，即平台创新、玩法创新及营销创新。对于《捕鱼达人》，同样存在这三点创新。

《捕鱼达人》的前身是安装在游戏机上的一款中型游戏，我们最先把它迁移到iPad这个新平台上。新平台的选择本身就是一种创新，在该平台市场启动之前即可首先获得快速发展的优势。在玩法上，我们并不是一味地照搬原有游戏的玩法，而是加入了我们自创的一些玩法，用赠送金币的方法激励用户。可以说现在的《捕鱼达人》和其前身在游戏概念上已经完全不同了。至于营销，我们希望能像《愤怒的小鸟》一样推出一系列相关的衍生品，比如希望有乐队制作以《捕鱼达人》为主题的音乐，在乐淘上推出《捕鱼达人》的鞋子等。



陈昊芝表示，要用iOS思想去做创新产品，而不是简单山寨

争显中国创造精神

我们将全国95%的用户定位为《捕鱼达人》的目标用户。由于中国3G网络还不稳定，我们将联网标准确定为GPRS，这样所有用户就可以在顺畅的网络环境中玩这款休闲游戏。可以说对于任何一款休闲游戏，都将有全国95%的用户群，发展空间是极其大的。同时游戏中内置计费的收入，也将为公司带来可观的收益。

在拥有巨大发展机遇的同时，中小型开发团队也面临着来自国内及国际巨头的竞争压力。在竞争实力上，小型开发者并不占据优势，但也不必对这个市场过度灰心。只要国内开发者能集中精力联合起来，用做iOS产品的思维去思考创新，而不是简单的山寨；能相互之间彼此支撑，共同遵守海外真正的市场规则，相信终有一天，我们可以很自豪地站在国际市场上说“我们是中国开发团队”。在这一点上我们需要向中央美院柠檬岛工作室学习，他们开发的每一款儿童应用几乎都被苹果在首屏进行推荐。

InfoLife团队：

注重用户反馈迅速做出应对

记者 / 杨鹏飞

InfoLife总部位于美国，目前开发了二十多款Android应用，其中最受欢迎的三款为：高级任务管理器（Advanced Task Manager）、历史记录清除器（History Eraser）和轻松卸载（Easy Uninstaller），在国内Android用户中有很高的知名度。近日，《程序员》采访了InfoLife团队，请他们分享创业的经验。

《程序员》：InfoLife团队是怎样组建的？

InfoLife：发起人是陈琛，他原来在启明星辰，尝试开发过一款音乐播放器。发现效果不错，于是就联系我们，想一起做事业。后来，这个播放器被我们放弃了，因为我们只是想做一个实验，尝试Android平台，也没确定要做下去。

《程序员》：当初为何想到要做Android应用？在这之前你们有开发游戏的经验吗？有反馈指出你们的界面并不突出，有没有考虑换掉现在的图标？

InfoLife：2009年6月左右，我们开始做InfoLife。之前我们已经尝试过各种方向，比如小游戏。这期间，我们发现小游戏用户的依赖程度不够，用户黏性相对较差，纯粹从盈利角度来看不是很理想，用户的下载量都没有达到我们的预期。再加上当时硬件配置不高，另我们很难继续下去。因此，我们放弃了小游戏，转向开发购物类应用。

由于对北美和欧洲市场的实际情况不太了解，做出来的购物类应用不太符合用户口味，于是两个多月后，再次决定放弃，开始做系统应用，也就是InfoLife。从2009年8月底开始做，10月1日正式上线，前三个月达到25万下载量，相当于到了一个市场的最高级别。我们比较欠缺游戏开发的经验，同时我们也缺少精通用户界面设计的人员。用户界面十分重要，但我们四个人都是做开发的，并不擅长这方面的工作。现在的图标已经被用户接受，突然更换可能让用户不适应。但在新应用中我们会把用户界面做好。

《程序员》：你们成功的关键有哪些？如何找到正确的方向？

InfoLife：一开始就要找对方向，还有就是专注、投入，一旦做起来就坚持下去。我们非常注重用户的反馈，通过邮件和Android Market上回复用户的评价这两种方式。坚持尽量全部回复用户的反馈，对用户需求做快速反应。我们几个人每天都会收到几百封邮件，一般只要是能看得懂的语言、自己能翻译得好的语言，我们都会回复，同时不断修正Bug。

2010年Android手机出货量暴涨，我们的应用下载量也随之迅速增长。然后通过用户的反馈，了解下载量增长的原因。通过友盟的统计，对用户黏性进行一定的分析。

《程序员》：InfoLife团队现在一共开发了多少款应用？有没有分析造成这些产品下载量高低的差异？

InfoLife：我们一共做了二十几款应用，差异主要跟定位有关系。有的可能比较适合高端用户，而有些普通用户就可以直接上手使

用，所以下载量会有区别。

《程序员》：有涉及到HTML5开发吗？哪个因素对产品的下载量影响比较大？有没有担心未来Android会将你们应用实现的功能集成到Android系统中去？

InfoLife：HTML5比较少，主要还是在Android平台，感觉HTML5的市场现在还不成熟，我们目前就是坚持做好产品，还不会考虑跨平台的问题。

移动互联网市场现在尚未形成一个固定的格局，会有很多变数。对于Google官方系统集成的问题，我们也有过这样的担心。这个市场变化非常大，所以各种调整要非常快。影响最大的就是Android Market的算法排名。用户评级是硬指标，还有就是短时期内的下载量。它是有一个周期，然后下载量满足要求就会排名靠前。但长期的话还是看产品本身的品质，如果应用再怎么调整，排名变化不大，那么这个产品就算是做到位了。

我们认为云计算很重要，移动互联网的未来就是App加云这种模式，所以我们现在急需资源来把云做大，把用户的数据保护好。

《程序员》：你们现在月收入是什么情况？如何做推广？

InfoLife：主要收入来自广告和收费，广告占到70%。主要收入分布在3～5个应用里面。其中北美最多，占到60%多；第二是澳大利亚，占到30%；欧洲大概占10%左右。我们目前主要是做海外市场，不过已经开始准备进军国内市场。国内现在也有很多“我们”的产品，但都不是我们发布的。未来我觉得国内还是很好的。市场量更大，做什么都有前景。

至于推广，我们主要是靠口碑传播，比如国外有个知名网站AppBrain【注：AppBrain功能包括根据本地安装的应用进行应用推荐、查看好友安装的应用、热榜应用，还支持直接用Google帐号登录，在网站上标记需要安装的应用，用户可以随后在手机上sync安装，不用通过手机在市场里面找应用下载安装】就愿意给我们做推荐。也没有专门去做SNS营销，比



如Facebook、Twitter一类。同时我们也尝试过各种广告形式，最后发现AdMob效果最好，Android Market本身还是最有效的推广途径。

《程序员》：Android版本更新很快，对你们影响大吗？你们的产品支持平板吗？

InfoLife：有一定影响，比如在一些细节方面如果不注意就会产生Bug。Android 3.0的改变比较大，用户体验也很不一样，需要对应用做一些调整。我们更多的精力都是放在新应用上，一些老的应用需求相对稳定，主要是进行Bug修订的工作。

InfoLife在平板上可以运行，但我们还没有针对平板的用户体验做进一步的优化。主要还是平板的量很少，所以就没怎么关注，而且系统也不够完善。

《程序员》：对一个没什么经验的年轻团队，也想做Android方面，有什么建议？

InfoLife：我的建议就是快速行动，先干起来，初期可以不要投入太多。App其实算互联网产品，不算传统软件，所以需要敏捷开发、快速迭代。一个应用火不火，体现在它的下载量上，每天下载量至少达到3000以上就是一款很有前景的应用。而我们的下载量最高将近2万，总下载量达到了1200多万。

天天动听CEO黄晓杰：

移动互联网现在谈盈利还太早

记者 / 陈博

历时四年，从一无所有到拥有5000万的下载量，在外人看来，天天动听的开发团队已经取得了巨大的成功。而其CEO黄晓杰却认为，在当前的移动互联网环境下，天天动听离成功还有很长的路要走。

软件研发模式需要升级

《程序员》：请简要介绍你们的团队。你的主要工作有哪些？

黄晓杰：我和我的两位搭档——赵然和王智罡，已经风雨同舟、合作了四年之久。王智罡是我老乡，在老乡会上认识，相谈甚欢，因为我们都想做大事。虽然当时我们一无所有，但都觉得房子、车子这些不是我们的追求，我们想要做成一件能够影响更多人的事。后来我又邀请了一位志同道合的老同事赵然加入，于是仅靠着我们这三杆枪就开始了创业之路。我现在主要负责产品和战略，赵然负责研发，王智罡负责商务和运营。

《程序员》：在天天动听的整个开发流程中，哪个环节的挑战最大？

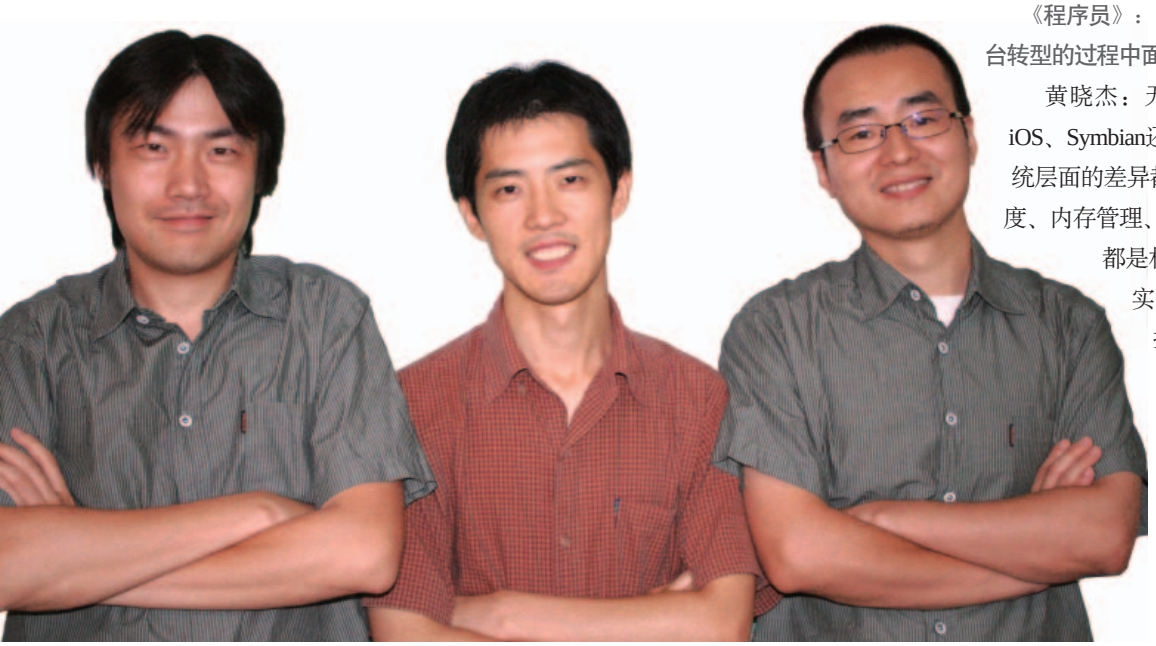
黄晓杰：现在我们面临的最大挑战在于如何保证可持续的快速迭代。在公司创立初期只需靠能力强的人员便可达成这个目标，但随着公司规模扩大，人员素质参差不齐，代码质量严重下滑，我想这是每个软件公司都会遇到的问题。于是，我们查阅了大量软件工程资料，发现过去我们的开发模式是手工作坊型的旧模式，现在需要向正规的软件研发体系升级。其实行业内部早就有较为成熟的模式，譬如阿朱提到的“三驾马车”以及最近流行的敏捷开发等。

智能平台用户更关注体验

《程序员》：从Symbian平台向Android平台转型的过程中面临哪些挑战？

黄晓杰：无论是Windows、Linux、iOS、Symbian还是Android，它们在操作系统层面的差异都不大，进程管理、线程调度、内存管理、文件I/O、虚拟内存等理念都是相同的。对于一个基本功扎实、充分理解操作系统、数据结构的程序员，切换平台至多需要两个月，剩下的工作是不断了解平台特性，其实门槛并不高。

Symbian和Android的终端差异很大。首先，其用户习惯不同，一个是键盘机，一个是触摸机；其次，用户群体



天天动听创始团队从左至右：王智罡、赵然、黄晓杰

差异很大，一个是中低端用户，另一个是中高端用户；再次，用户关注点也有很大差异，一个非常在意流量，一个不太在意流量而更关注体验。因此，在产品设计上，需要做出很大的变革。

《程序员》：天天动听刚刚发布iOS平台的产品，有哪些新特色？

黄晓杰：iOS版本上线三天就成为App Store音乐类排行榜第一，说明用户对我们的产品非常期待，虽然第一个版本只拥有最基本的功能，但还是获得了用户的好评和鼓励。1.0版本没有太多亮点，我们的后续版本会不断带给大家惊喜。

唯一的考量对象是用户

《程序员》：在移动应用的市场推广和发布方面，天天动听面临的主要挑战有哪些？

黄晓杰：天天动听起步较早，所以现在形成了强大的品牌效应，在推广和发布上不用花太多心思。如果未来竞争进入完全白热化，产品严重同质化，那时渠道的作用就会非常重要。在Android平台，应用商店成为非常重要的软件发行渠道。天天动听在国内大部分应用商店已经上架，我们和应用商店形成了互利共赢的合作关系：一方面，我们的产品非常受用户欢迎，能够为应用商店提高活跃度；另一方面，应用商店能够帮助天天动听获得更广泛的用户。

《程序员》：天天动听如何在与手机QQ音乐这种大厂商的产品竞争中占据优势？

黄晓杰：其实创业公司最大的对手是自己，最需要考量的对象是用户。如果每天非要找一个靶子来练，很容易走火入魔。我个人非常讨厌阴谋论，不需要把一件非常简单的事情复杂化，只要团队的能力提高了，把用户体验做好了，用户自然会选择你。

目前谈盈利还太早

《程序员》：天天动听目前已有5000万装机量，你们暂不考虑商业模式的原因是什么？你认为小型团队的盈利之道在哪儿？

黄晓杰：移动互联网还处于非常初期的阶段，上网设备、带宽、资费、支付等环节的很多问题都没有得到解决，所以我认为现在谈盈利还太早。PC互联网上非常成熟的三种盈利模式——广告、网游和电子商务，在移动互联网都还未崛起，能够盈利的公司寥寥无几。同时，PC互联网的传统势力都已一头扎进了移动互联网，过去在手机上形成的内容和品牌优势很容易被冲垮，所以在移动互联网上已经取得一些成绩的团队需要格外小心。小团队可以赚点小钱，但千万别忘记互联网的本质是流量，移动互联网还有很长的路要走。要想活着看到后天早上的太阳，就必须足够专注，找到可以低成本持续获得用户量和流量的方法。

《程序员》：面对目前的移动互联网形势，你对应用开发者有何建议？

黄晓杰：现在行业正处于窄带移动互联网向宽带移动互联网转型的阶段。虽然这是个大趋势，但要把握好节奏。开发者不要期望3G+Wi-Fi的时代一天就能到来，过去很多人就是因为太乐观，犯了很严重的错误。

《程序员》：从一无所有到天天动听达到5000万的装机量，你能否为刚起步的小型创业者分享一些经验？

黄晓杰：其实到现在为止，天天动听离成功还很远，不敢妄谈经验。我们团队非常认同雷军所说的“专注、快速、极致、口碑”的理念。人间正道是沧桑，如果想做成一件伟大的事业，就必须能够耐得住寂寞、抵得住诱惑。成功的榜样非常多，关键是要虚心学习。我建议大家把互联网大佬的视频都看一遍，相信会从中得到很多启发。

记者后记：虽然还未走上盈利之道，但天天动听团队对产品开发的心无旁骛、对移动互联网形式的冷静判断、团队自身的内敛谦逊都是开发者学习的典范。希望每位开发者都能专心做好产品、认清形势、步步为营，因为机会总是留给有准备的人。

科大讯飞副总裁江涛：

移动互联网时代的语音识别技术

记者 / 杨东杰

语音识别技术已成为移动终端上最受关注的交互技术之一。由科大讯飞公司开发的讯飞语音输入法自推出以来，已拥有数百万Android用户，其iOS版本也表现不俗，产品背后是科大讯飞国际领先的语音识别技术，去年推出的语音云平台也受到移动应用开发者的关注。

“讯飞语音输入法能判断语气吗？”台下有人问正在演讲的科大讯飞副总裁江涛，江涛并没有直接回答，而是对着自己的手机说了一句“你今天和我一起吃饭吗？”和手机同步的大屏幕上马上准确显示出这句话的文字，然后奇妙地自动附上了问号“？”，现场掀起一片掌声。这是7月14日，在CSDN TUP人机交互主题活动上的一幕。

讯飞语音输入法是由科大讯飞公司推出的一款新型输入法，基于科大语音云技术，将用户语音输入的日常用语快速转换为文字，其识别率令人惊喜。上线以来，其Android版本已拥有数百万用户，iOS版本（口讯）在今年5月曾一度登上App Store中国区榜首，得到近1000个评价，其中85%用户给予其五星级评分。

输入法背后的智能语音技术

语音技术已成为移动终端上最受关注的交互技术之一，今年8月份，有消息称即将推出的iOS 5操作系统将继承语音识别功能，如果消息如实，将大力推动语音识别技术的应用发展。消息中，苹果公司在语音领域的合作伙伴是著名的语音和图像解决方案提供商Nuance。

Nuance公司（Nuance Communications, Inc.）是全球最大的专门从事语音识别软件、图像处理软件及输入法软件研发、销售的公司。目前世界上最先进的电脑语音识别软件Naturally Speaking就出自于Nuance公司。

智能语音技术，通俗的说，就是使信息时代的各种信息机器像人一样“能听会说”的技术，包括语音合成、语音识别和语音评测等。江涛在演讲中，对此作了具体介绍：

- 语音合成技术可以将任意的文字信息转化为自然流畅的语音，相当于给机器装上了人工嘴巴；
- 语音识别技术可以将语音中内容、说话人、语种等信息识别出来，相当于给机器装上了人工耳朵；
- 语音评测技术可以进行发音标准评价和错误反馈指导，相当于把机器变成语音评测老师；

科大讯飞可被视作中国的“Nuance”，目前已是中国最大的智能语音技术提供商。据科大讯飞副总裁江涛介绍，核心技术方面，科大讯飞在语音合成、语音识别、语音评测、声纹识别等语音相关的领域都保持了国际领先；产业化方面，讯飞作为国内唯一的语音技术上市公司，目前公司有1000多名员工，市值在100亿左右。

云计算和移动互联网时代下的语音识别技术

除非是语音技术从业人员，可能现在很少有人记得，1999年IBM曾隆重地推出一款叫做ViaVoice的连续语音识别软件，使用者可以通过语音进行文字输入，在当时引起了巨大的轰动，成为当年全球IT界十件大事之一，可惜这款产品最后并没有获得市场的认可。

1999年，也正是科大讯飞的几位核心创始人刚走出校园、迈向产业的时候，他们深入地

分析过ViaVoice失败的原因，最终的结论是“生不逢时”——首先，ViaVoice是用在PC上的，多年来PC上鼠标+键盘的输入方式已经非常高效而且精确，语音输入在PC上给用户带来的价值提升有限；其次，单机软件还是当时软件的主流模式，这种模式相对当前的在线服务模式，在持续训练、迭代优化方面都有很大的局限性。

进入移动互联网时代后，以手机为代表的移动互联网终端普遍存在输入不方便的问题，语音输入在便捷性方面的优势极大地凸显出来；另外，无线带宽的提升和云计算技术体系的成熟，手机上的复杂运算可以交给后台，这样可以通过迭代优化的模式不断训练语言模型、声学模型，语音识别的准确率能够持续提升。现在可以肯定的说，移动互联网时代的到来，也是语音技术应用春天的到来。

江涛认为，语音技术在移动应用中，有三个方向比较被看好。第一，效率提升。在应用中集成语音接口使用户可以快速方便的输入，例如在微博、记事本等应用中集成语音输入，可以改善用户的使用体验。第二，教育学习。语音是语言学习的重要部分，语音交互技术用在帮助学习方面，效果显著。第三，娱乐。Talking Tom只是利用一个非常简单的语音回放变调技术，就能够风靡全球，说明语音在娱乐方向可以挖掘的点很多，可以做出更多有趣的应用。

面向开发者的讯飞语音云平台

之前科大讯飞主要面向行业厂商，为电信、金融和教育厂商服务。讯飞语音输入法是公司进军大众消费级市场的一个试点，去年10月和输入法一起推出的，还有讯飞自主研发的全球首个同时提供语音合成、语音搜索、语音听写等智能语音交互能力的移动互联网智能交互平台，基于该平台，众多中小开发者可以低门槛获得语音交互能力，开发出自己的语音应用。

目前讯飞语音云上面提供语音合成、语音搜索或者说语音识别等接口，未来将进一步开放声纹识别等功能。



科大讯飞副总裁江涛认为，移动互联网是语音识别技术的真正用武之地

讯飞语音云为开发者提供两种接口：第一，C风格的API接口，可实现最广泛的平台覆盖性；第二，可视化控件接口，为了提高智能终端开发效率、降低开发难度，为Android、iPhone、Symbian平台提供了内置语音交互UI、录音放音封装的可视化语音控件接口。

开发者可以选择适合自己的合作模式，在初创期间可免费使用，达到一定量之后，可以选择流量交换或技术服务费等形式进行分成。现已有50多个应用使用了讯飞语音云接口，有超过500个注册开发者。

而讯飞语音输入法就是讯飞语音云的示范应用，实际测试中，普通人电脑标准键盘上输入速度是每分钟50~80个字，在iPhone等触屏类型的智能手机上只能达到15~20个字，而语音输入每分钟可以达到200字以上。未来随着语音识别准确率的持续提升，语音交互和语音输入必将成为手机等移动终端最重要的人机交互界面之一。

RockPlayer创始人张永强：

团结就是力量

记者 / 杨鹏飞

RockPlayer是一款嵌入式平台上的高性能全格式视频播放程序，有着专为移动设备优化的播放核心以及高度优化的FFmpeg解码器。张永强是RockPlayer创始人，他认为是团队成员之间的信任、团结和良好的相互沟通，造就了今天RockPlayer的良好口碑。当然良好的眼光，看准方向也很重要。

《程序员》：你们是什么时候决定开发RockPlayer的？起因是什么？

张永强：早在2007年时，我就想开发这样一个视频播放器了。当时视频和音频内容已经开始日渐流行，而手机播放视频文件的能力非常有限，尤其是很多PC上可以看的视频格式，在手机上无法正常播放。虽然很多厂商随机提供了一些视频格式转换软件，但这些软件需要安装在PC平台才能使用，且转换过程漫长，导致手机上看视频非常麻烦。所以那时我就认为，手机视频播放器这个市场确实存在，而且很大。我们看准了这个市场，才开始着手开发RockPlayer。

《程序员》：做播放器软件最关键的地方是什么？移动设备的屏幕分辨率和尺寸对软件的影响有多大？

张永强：主要就是在各个环节进行优化，从核心到UI都要进行优化。一定要充分利用设备的能力，将设备的硬件充分利用起来，才能做出高性能、高质量的播放器。性能是一款播放器最重要的评价参数，高性能的播放器，用户用起来才会觉得好。

分辨率的碎片化，带来的影响主要还是在UI方面。我们需要根据不同屏幕的分辨率进行适配。而对播放器程序本身的性能倒是影响不大。

《程序员》：开发过程中，最让人崩溃的事情是什么？而最让你觉得开心的是什么？

张永强：最让我崩溃的事情还是来自盗版方面带来的困扰。我们的免费版本是带有广告的，但去广告免激活完全版本在我们发布几个小时以后就出现了，当时确实比较崩溃。

说到最开心的事情，当然就是我们做出来的产品确确实实地解决了用户在播放视频文件中，碰到的问题。

《程序员》：你们是怎样做推广的？现在的月下载量和总用户量有多少？

张永强：推广主要靠口碑，基本没做过其他形式的营销推广。

现在每个月下载量有几十万，总的用户量达到百万级别。

《程序员》：你们团队是什么时候成立的？总共多少人？团队刚成立的时候，有没有遇到困难时期？

张永强：我们是2009年成立的，现在总共有4个人，其中3个是做开发的。

困难时期当然有。那时因为大家对产品和团队的未来认识不统一，各有分歧，公司产生了很严重的危机，甚至有人真的选择了离开。好在现在的团队成员之间相互非常团结，目标一致，多年来建立起来的信任让我们彼此都有很好的沟通。所以RockPlayer坚持走到了今天，并获得广大用户的认可。

《程序员》：你最得意的作品是什么？

张永强：我觉得最得意的，可能不是具体的某个作品，而是整个团队对技术最前沿的不懈追求。

Google商业合作部赵伊江：

应用开发者如何善用广告盈利

记者 / 杨东杰

移动广告已成为应用开发者主流盈利模式之一，赵伊江指出，已有一批中国开发者通过AdMob年达到上千万元，他建议应用开发者在产品设计之初就要考虑到移动广告的设置和运营。

移动互联网市场至少是传统互联网的十倍？这种说法还有待验证，但移动互联网正在快速发展速度是有目共睹的事实。

如果将2009年视作中国移动互联网的元年，那么只有2岁的它现在已经进入了激烈的市场竞争阶段。

以移动广告为例，据不完全统计，仅在中国，就有超过40家的第三方移动广告厂商，每家企业都将做“中国的AdMob”视为自己的奋斗目标。另一个有趣的事实是，由于填充率不能令开发者满意，提供广告聚合业务的厂商就有三家。

进入2011年，更多的开发者开始关注应用的运营推广和盈利模式。而在免费模式盛行的中国市场，广告成为最重要的盈利方式之一。近日，《程序员》专访了Google商业合作部赵伊江，了解移动广告的最新发展情况及其对开发者的价值。

应用规模在移动广告上的价值体现

《程序员》：移动广告对开发者盈利的价值，在AdMob上有没有一些参考性的数据和案例？

赵伊江：去年就有这样标杆性的数据，通过AdMob全球移动广告网络，已有一批中国的应用开发者获得千万级的人民币收入。

移动广告帮助开发者盈利的案例很多，例如我们的一个合作伙伴——Android平台上优秀的游戏开发商卓亨公司，在他们的创业早期，75%的盈利来自于Google AdMob。

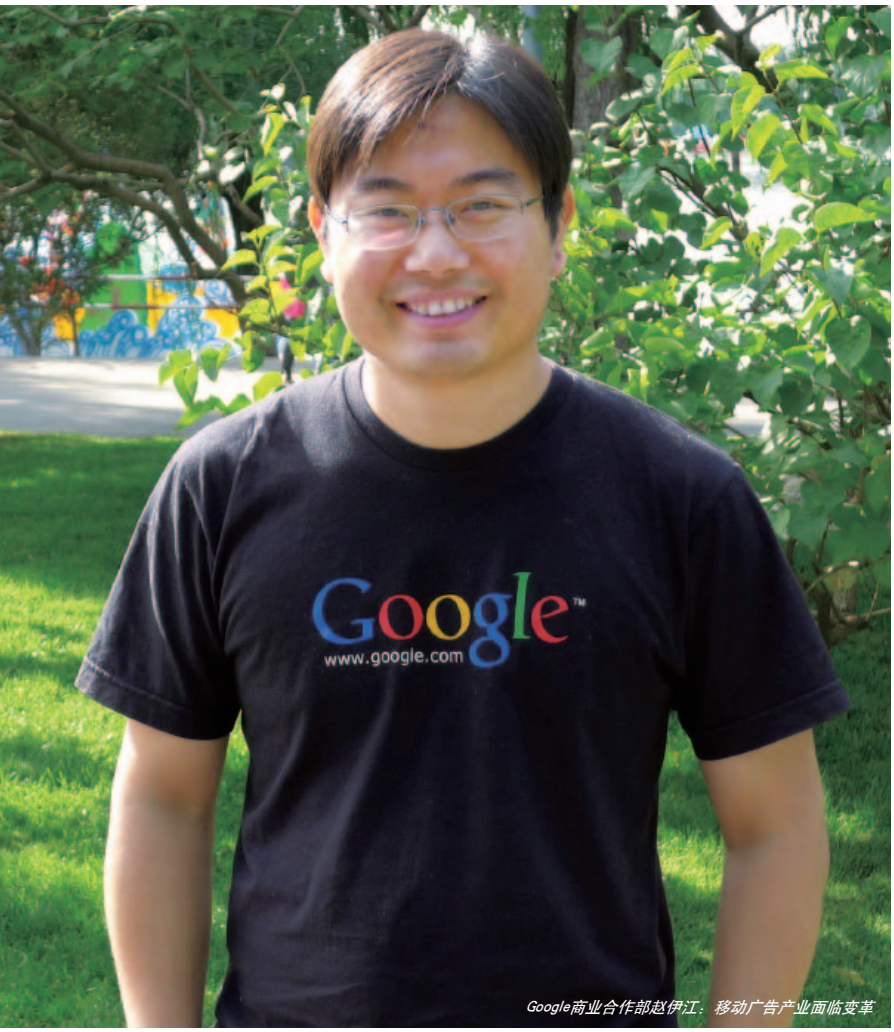
另外，还有一个关键数据，根据我们去年第三季度移动广告上收入的年度推算率，Google 2010年在移动广告领域的收入已经达到了10亿美元（年度推算率是广告网络以本季度收入乘以4，用于预测年收入的数据单元）。这个收入不仅仅来自移动应用或者移动网站的展示广告，也包括手机搜索广告。

《程序员》：如果应用达到一千万下载量，那么应用的价值如何在移动广告上体现？

赵伊江：展示广告是目前主要的形式，其收入计算公式为：**移动广告收入=页面浏览量×点击率×单价**。在页面浏览量（PV，Page View）、下载量和广告展示量之间有一个转化的过程。要将激活比例、每个用户每天打开应用的比例、每次产生的PV、PV中含有广告的比例这些因素考虑进来。

以一千万的下载量为例，如果激活比例为80%，20%用户每天浏览，每个用户每天浏览1次，每次产生PV为10，PV中含有广告的比例为50%，那么这个应用就得到800万的PV。得到PV后，还不能简单地计算出应用一天可以赚多少钱，原因在于还需考虑点击率和广告单价。一个好现象是移动互联网上的点击率要远远高于互联网，因为移动广告更显眼，而且现在有很多创新的移动广告形式能够使用户更有兴趣进行互动。

点击率和广告匹配以及广告位置有关。广告匹配主要基于选择移动广告网络的匹配技术



Google商业合作部赵伊江：移动广告产业面临变革

与应用的相关度。相关度越高，点击率越高。对于广告位置的设计，如果开发者希望将广告盈利作为移动应用主要的收入来源，那么建议在产品设计初期就考虑广告模式的嵌入，而不要等产品开发完成后才思考该把广告放在什么位置。

对于单价的制订，广告主规模、应用所在的行业、季节性和应用对广告主的价值是主要影响因素。一般来讲，利润率高、广告主多、而高质量应用少的行业单价会相对较高，所以做有内容有特色的应用很重要。比如汽车或保险类广告主就会比较青睐汽车车主常使用的应用。还有季节性的问题，广告主在不同时间点会有不同投放行为。比如在节日期间，广告主会集中投放，造成单价的上升。

最后也是最重要的是应用对广告主的价值，用户在应用中点击了广告，能否给广告主带来收益是非常重要的。所以开发者要注重提高应用质量，避免诱导用户点击的行为发生。诱导点击不仅可能造成帐户的停用，而且由于没有为广告主带来价值，也会影响单价。

因此，如果开发者想通过移动广告来盈利，要注重结合应用特点提升用户黏性和活跃度，注重内容质量建设。此外，在尽可能少地影响体验的前提下不要将广告隐藏太深，如在游戏中善于利用自然间隙插入广告。

移动广告盈利的四大要素

《程序员》：中国移动互联网市场和海外市场存在巨大差异，请介绍AdMob在中国的主要工作？

赵伊江：AdMob在中国发展很快，去年12月份AdMob中国的月广告请求超过10亿。今年这个数据会有很大增加。

目前我们的主要工作是和总部保持密切联系，每周都会进行沟通，以此保证中国开发者的需求可以得到快速响应和优先考虑。为此，我们特别在商业合作部建立了针对移动业务的团队。

我们也不断组织和参与行业活动和论坛，比如前不久Google就独家赞助了CocoaChina组织的第三届中国移动互联网iOS开发者大会，致力于为移动开发者建立一个沟通和交流平台，并将移动开发、移动广告的最新趋势介绍给中国开发者。

《程序员》：面对国内数十家移动广告的竞争，AdMob有哪些优势和价值？

赵伊江：移动应用开发者在选择广告平台时，一般会考虑几个因素。

首先是广告网络的规模，包括广告主的规模以及发布商的规模。从广告主来讲，AdMob被Google收购前作为全球最大的移动广告网络

就已经积累了众多移动行业的广告主，收购后Google全球数以百万计的广告主中很多家都已经成为AdMob的广告主，这也是一直来AdMob填充率高的原因。从发布商规模的角度来看，全球超过8万个应用在使用AdMob，覆盖主要运营商和设备，71%的Android和iOS用户到达率，到2011年5月每天的广告请求数已超过27亿，这样的广告网络规模对广告主非常有吸引力。

其次，AdMob技术也达到全球领先水平，可以真正做到精准广告投放。因为移动媒体上的内容相比互联网较少，所以广告匹配的技术门槛较高。此外，AdMob在2007年成立后在移动广告技术领域就长期处于领先地位，也是全球最早支持在移动设备投放富媒体广告的广告网络。目前，AdMob支持多种广告样式，包括文字、图片、富媒体、扩展广告、插播视频、HTML5等，广告主可以在此基础上制作出丰富多样的广告形式。

《程序员》：对于更好地利用AdMob广告盈利，你有何建议？

赵伊江：对于开发者来说，首先应遵守总的原则——做优秀的应用吸引有价值的人群，进而吸引精彩广告的投放。

在应用定型的情况下，一些小技巧可以帮助开发者最大限度的利用移动广告。众所周知，流量、点击率、单价、填充率是影响广告盈利的4大关键要素。针对这些要素有如下建议。

- 提升广告展示量。重视设置页、得分页等页面；将弹出窗口改为静态页面（弹出窗口无法放置广告）；采取弹出窗口等创新的方式推广新应用，从而降低交叉推广自家应用所占广告位的比例。
- 提升填充率。强烈建议开发者开通AdSense补余。
- 提升点击率。测试广告放在屏幕不同位置的不同效果；改变文字广告，文字和背景的颜色；在产品设计初期即考虑广告位的设置；在后台为应用选择合适的类别；设置合适的刷新频率。

移动广告最大挑战在于投放滞后

《程序员》：目前移动广告发展遇到的最大挑战是什么？

赵伊江：一方面移动流量高速增长，另一方面移动广告增长相对滞后，这就是很多开发者抱怨的填充率不足问题的来源。根据2010年的调查，美国最大的在线广告主中，有69%没有针对移动网站进行投放，也就是说很多广告主也还没有做好投放移动广告的准备。

不过，这个情况正在发生变化。今年，Google与IPSOS市场研究公司合作，进行了手机用户行为的调研，调查表明中国的受访者每三人中就有一人拥有至少两部手机，这个比例在亚洲是最高的。中国用户在手机里安装的应用程序平均数量为15个。中国城市居民拥有智能手机超过一年以上的人数比率是最高的。

随着用户注意力从PC互联网到移动互联网的转移加速，先转向移动网络的广告主取得显著效果，广告行业将迅速做出反应。

《程序员》：前段时间，有消息称，AdMob上针对移动Web的广告将分离到AdSense上，可否透露背后的原因？

赵伊江：AdSense和AdMob在各自不同的领域都有一些品牌和技术的积累，我们实际上是为了帮助广告主和发布商更好地实现共赢才作出了一个选择。

最初，AdMob已经看到将来移动网站和移动应用上的广告前景会非常大，所以做了两个都可以用到的解决方案。但在收购之后，我们觉得AdSense在移动网页上更有优势，所以未来AdMob就不会支持移动网页上的广告，而只支持应用里的广告。

面向开发者的一端是更清晰的划分和定位，即移动网站使用AdSense，移动应用使用AdMob，对开发者的支持工作统一由Google商业合作部承担。在Google，我们正在根据剧烈变革进行技术升级，未来Google Adsense、AdMob、DoubleClick™将走向融合。

“建立企业中的用户体验力量”系列之二

用户体验质量控制体系

文 / 吴卓浩

用户体验质量的评判要素有哪些？最容易出现哪方面的问题？本文为您讲解如何建立用户体验的质量控制体系。

许多刚开始接触用户体验概率的企业非常希望能有一套标准体系，照做就可以保证产品的优质用户体验。其实，有许多讲解用户体验评估要素和方法的公开资源，那么为什么还是只有少数产品拥有优质的用户体验呢？这其中有什么“秘密”？

用户体验质量的基础要素

■ **有用性**。满足用户的需求，为用户解决实际问题，给用户带来价值。比如开发者需要明确输入法产品的主要功用是帮助用户在手机上进行更快的文字输入，而不是在手机上进行文字输入的同时利用手势动作给手机充电。当然，产品不一定只能围绕用户的现有需求。事实上很多优秀的产品就是通过创造性地发掘用户需求并解决，从而为产品带来巨大价值。比如，直观的触摸控制、随时随地连接网络、利用随身携带的移动设备，做以往需要通过不同设备才能实现的事情，当这些用户需求被创造性地发掘、整合和满足时，就出现了iPhone这样划时代的产品。

■ **可用性/易用性**。这是用户体验中最核心也是最庞杂的部分，包括可识别、可理解、可预测、掌控感、及时反馈、操作连续、一致性、稳定、可靠、效率、使用疲劳度、易记忆、避免用户犯错并宽容对待用户错误等内容。可用性不仅包括界面设计，也包括产品的整体表现。比如产品是否使用流畅、稳定，就是典型的涉及产品整体表现的例子。另外，在不同的使用情景下，可用性的侧重点也不同。比如坐在办公室中用手机拨打电话和开车时用手机拨打电话，对产品可用性的要求就不同，相应的对产品的要求也不同。

■ **感性满意度**：给予用户美感、愉悦或其他特定感受（比如游戏中的紧张感），给予用户成就感或被关注、被尊重等感受。这些感受甚至可以延伸到产品之外，比如让用户觉得自己使用某个产品，在朋友圈里就显得很牛。

■ **激发性**。经典的用户体验评估主要包括以上三种要素，但随着Web 2.0、社交网络的兴起，在融入了分享和社交元素的产品以及很多游戏和探索创造型产品（比如绘图、音乐创作等）中，可以引入一个新的基础要素，用来评估产品激发用户进行探索、创造、分享、互动的活跃程度。用户的这些行为已经超越了单纯的把产品作为工具使用的概念，而是把产品作为一个平台，发挥自己的能力和潜力，和其他用户一起产生聚变的力量。营造这种用户体验的要素，我称之为激发性。

实际产品的常见问题

在实际产品中，最容易判断却又最常见的是和可用性有关的问题。

■ **缺乏重点和引导**。用户进入页面，看到的东西很多，却不知道从何入手、该做什么，或者重要的功能被隐藏起来让用户很难发现。

■ **缺乏明确的规则和一致性**。如一个界面上的部分文字或图案可以点击、另一部分文字或图案不能点击；有些操作自动保存、有些操作默认不保存；同一个功能的按钮在不同界面上位置不同、或者形态不同。

■ **界面设计容易引起误操作**。当用户发现时已造成了严重后果，或者让用户产生困惑。

其次，经常会出现和感性满意度相关的问题。

■ **犯基础设计错误**。对界面元素的色彩、段落的排布、字体的选取过于随意，是最容易让产品显得不专业的几个方面。

■ 采**用与产品气质不符的视觉设计**。把轻量的网络应用设计成层次复杂、质感厚重，或者在精致的界面中突然出现一个简陋的系统自带控件，以及抄袭其他知名产品的设计。

■ **为形式而形式**。故意做一些很酷的效果，而不是为了更好地实现功能，甚至影响正常操作。

然而，并不是仅仅依照这些用户体验要素定制产品就一定能获得优质的用户体验。首先，不同类型的产品需要不同的用户体验要素评价标准。比如工具类产品需要提高完成任务的效率，而游戏类产品反而可能需要降低完成任务的效率。面向普通用户的产品往往强调简洁易用，而企业级产品最重要的特质是稳定和可靠，反而要把操作做得复杂，反复确认以避免误操作；其次，即便是同一产品，在不同的使用情景下，可用性的侧重点也不同，需要全局化平衡考量；再次，在企业的不同发展阶段，需要制订相应的用户体验评价标准。比如有些产品在发展初期需要重点积累种子用户，用户体验侧重于让这些特殊的种子用户满意。而到了快速成长期，重点变为大量发展普通用户，用户体验就会变为兼顾普通用户和种子用户的需求。

要做好产品的用户体验质量控制，第一件事是想清楚产品究竟需要什么样的用户体验：怎样的用户，在何种情景下，用产品做什么；产品的用户体验近期目标和长期目标是什么。由此再去组织资源，制订工作流程，以及考评和激励手段。同时，用户体验的质量控制不能只靠评估。如果产品在研发过程中没有用户体验力量的参与，而在随后的评估中发现了问题，常常意味着大量的工作需要推倒重来，可往往已经没有时间进行修正了。

用户体验质量监控体系

用户体验质量控制体系不仅仅是产品完成时的检验，而应成为贯穿研发过程始终的工作。

■ **在研发过程中**，让用户体验工作尽早开始。进行访谈、焦点小组（Focus Group）、跟踪观察（Field Study或Diary Study）、问卷、数据分析等用户研究、基于用户体验的竞品分析，快速形成产品概念。



用户体验质量控制体系

■ **快速设计、快速检验**。充分进行设计探索，将设计想法做成效果图或原型，及早请目标用户进行测试，根据反馈快速改进方案。

■ **有条件的团队可以请资深的产品和用户体验人员组成委员会，定期开用户体验评审会**（UX/UI review）。有时，资浅但是不属于这个项目团队的产品人员或用户体验人员提出新鲜的意见和建议。同时，用户体验评审会也是资浅人员观察学习资深人员分析问题、讨论问题的好场所。

■ **请非项目组成员的用户体验人员详细试用和评估产品**（Cognitive Walkthrough）。

■ **在产品中预埋监测点**。等产品上线后收集用户数据，或者主动进行对比实验（A/B test），分析数据作为产品改进的依据。

■ **建立通畅的用户反馈渠道，以及鼓励用户反馈的机制**。

根据产品特质、用户特点、使用情景、产品发展阶段，选择相应方法、流程、评估要素，在研发过程中推动用户体验工作，就是质量控制体系保证用户体验质量的秘密。



吴卓浩

创新工场用户体验总监，曾创建Google在中国的用户体验团队。

责任编辑：陈博（chenbo@csdn.net）

手机社交游戏与触动用户的环节

文 / 郑金条

对于手机社交游戏，一般的进程权衡可能会包含三个层面的考量：开发者的修为、发行商的布局和玩家的生态环境。很多情况下，这三者之间的比重都显得不够协调，游戏开发层面往往占据了更大的分量，而忽略了玩家生态环境在其中扮演的关键角色。毕竟对于一款游戏，玩家市场才是最终敲定成败的主体环节。

开发者的修为

目前，手机游戏开发者由四个层面的人员构成：

第一，在手机游戏领域浸淫数年的老牌开发者和有成功案例的游戏新贵；

第二，由看中手机游戏未来市场方向的传统开发者所构建的团队；

第三，由其他领域人员跨行业搭建的游戏团队；

第四，一些独立游戏开发者。

其实在移动互联网崛起，手机游戏开发慢慢成为主流游戏市场认可的一部分后,手机游戏开发也演变成为新的利益角逐战。在这个新兴的产业中，大量开发者已经慢慢失去了做游戏初期的信念和理想。特别是一些临时拼凑的团队可能缺乏好的游戏思维，只是尽快通过山寨作品捕获用户进行套现。在低成本市场下，低门槛所造就的是更浮躁的行业氛围。所以提起开发者修为时，往往不敢直接鉴定是好的游戏理念带来好的游戏产品进而产生好的收益；还是快速切入、套现直接走资本的滚雪球效应。我们慢慢看不清楚究竟是谁在扮演着主体上的作用。

从Angry Birds崛起所带来的希望开始，手机游戏淘金战也如火如荼的展开，一些拥有好创意的产品率先成为行业的佼佼者。而因为跨平台和跨地区隔绝的关系，一些创意山寨游戏同样照葫芦画瓢取得了成功。甚至在部分市场，山寨往往能压过正品成为地区的主导。这也许是市场演进的正常过程，但也能从中更明显地折射出其和传统游戏产业的差异，游戏人的理念和心态已经被改观了。

发行商的布局

发行商是一个老生常谈的话题，说起发行商我们往往能够联想到两个不同的层面。

- 发行商真的有足够实力推动一款产品的成功吗?发行商真的有足够价值来瓜分走高比例的分成吗?发行商在持有众多产品的情况下真的能够像开发者一样对产品有足够的热情吗?
- 发行商拥有开发者所缺少的用户资源、媒介资源、平台资源和运营成熟度。可以有效抵御行业风险，让游戏更有出头机会。此外，在发行领域的专业素养会使得他们在操作上更娴熟，更能适应规则。

但无论在哪一个层面，都能映射出一个现实：在没有产品的情况下（特别是IP控制权），开发商在和开发者的角力中永远都会处在博弈状态，甚至会经常陷入培育了一款好产品转眼间却被单飞的窘境。

这个问题和一些游戏分享平台相似。有大用户量的开发者不愿贡献出自己的用户数据和

关系，而小型开发者在依赖分享平台的同时没有办法带来更多的注册分享用户。这种循环使分享平台注定会一直在和寻找机会的开发者打交道，而很难有更好的协作机会。

玩家的生态环境

玩家的生态环境问题是本文所着力论述的，就像前文所提到的，玩家才是游戏最终的生死场，开发者满意的产品如果不能在玩家市场获得有效回应也只能是开发者的单相思。

问题1：玩家文化构成对游戏题材接纳度的影响

一般开发者在敲定游戏题材时并不明确目标用户，即使有也是在后续修订中敲定的。除了通用型产品，游戏题材选择和玩家文化构成上休戚相关，包括玩家的文化习性（他们对所涉及题材的熟悉程度）、性别取向偏好、对游戏风格和画面形象的选择，以及所配置音乐是否吻合他们的听觉要求。更有甚者，会涉及部分信仰问题。

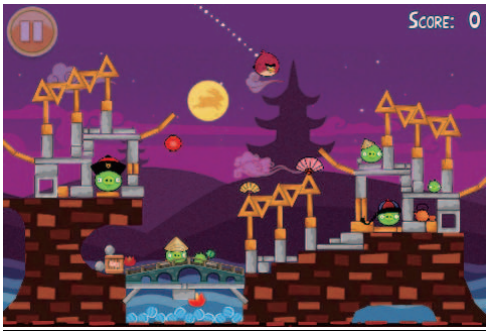
很多时候，开发者对待游戏本地化时所采用的只是简单翻译内容，有些游戏在国际化进程中的失败根源更在于其和当地文化甚至理念的格格不入，没有办法深入了解文化习性，从而带来理解方面的障碍。

尽管有些通用型产品本身没有这方面的顾虑，但并不是每一个创意都能预见到或探触到人类相似的共性灵魂和感觉，因此每一款游戏题材首先需要权衡游戏是否和玩家的娱乐需求有契合点。

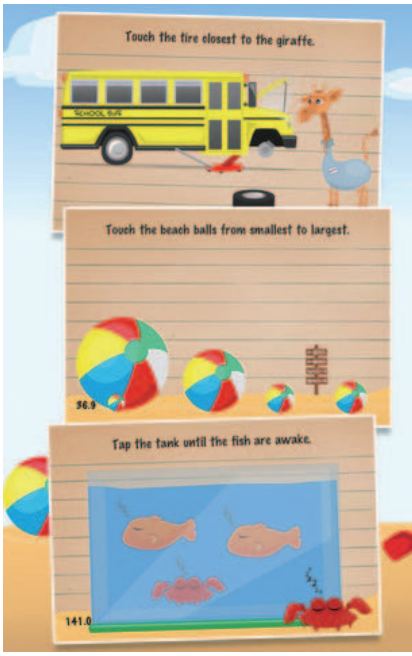
问题2：免费心理和玩家的忠诚度分析

在社交游戏和手机游戏领域我们常常在顾虑：为什么免费模式的游戏在用户留存率方面往往不尽如人意？

有三点分析值得思考：其一，付费游戏加强了游戏的附加值暗示，同时这种预先投入成本将会在潜意识中驾驭玩家，提示玩家在付出的情况下必然要寻求回报 ；其二，在免费游戏环节上，尽管时间成本可能是世界上最大的成本，但是玩家没有这种意识，免费意味着缺失牵制力或者增多了可选择性的风险；其三，因为制作相对简单，而开发者又想在新模式上有所突破，结果造成游戏性和交互性并没有被权衡清楚。







尽管存在不同层面的考量，但玩家才是游戏最终的生死场

在这种情况下，免费模式就需要驾驭在游戏本身的质量上，以避免用户的随意性态度。这里需要指出，由于游戏数量的增长和玩家对免费游戏本身价值感的降低，促使玩家并不能在游戏上做彻底的体验，造成了玩家流失风险的无限拔高。并且这种风险的把控并不完全在开发者手上，开发者所能做得就是在玩家的项目比照中脱颖而出。

问题3：迷茫的玩家和新手教程

所有玩家进入新游戏后都必须共同面对一个问题：这款游戏的乐趣在哪？该怎么玩？基本上游戏都会以新手教程的方式来引导玩家熟悉游戏中的相关流程，包括功能按钮的位置、大小和呈现方式；功能按钮介绍的繁复程度；是否真的能够一目了然的完成整体介绍。更为简单的说即让玩家在新手教程结束后能顺利进入游戏状态。

这一步对玩家的最终留存率至关重要，毕竟在现阶段，玩家活跃频度的价值远远高于单



开发者需要让规则本身在游戏进程中说话

纯的下载数据。

这就涉及游戏规则问题，若存在规则，玩家就得学习。有学习的存在就一定存在玩家对既定规则接受度的差异。开发者所能够做的就是尽快让玩家上手。并且需要应对一个情况：那些恼怒新手教程而直接跳过的玩家如何在游戏中继续接受有效地引导？毕竟玩家的自我探索可能更有趣，但同时存在让玩家不知所措的情况，多次的不知所措必然导致玩家的流失。

这就需要开发者明白游戏的完全规则，从而让规则本身在游戏进程中说话。让玩家向其他玩家解释游戏规则，并不会令过程变得更有趣。

问题4：开发者思维和玩家的心理目的权衡

我曾和不少开发者就开发者对玩家市场的自我预判和听取玩家建议两者的比值进行过交流。很多情况下，开发者并不能如Zynga那般通过对玩家测试，挖掘玩家真实需求，再进行快速迭代，大部分公司根据用户行为调整游戏方案一直停留在纯理论阶段。

现实的情况可能和Zynga所处的大格局并不相似，开发者的自我预判和对所获得反馈有所甄别。事实上除非有一定规模的数据参照，不然少量用户的反馈也可能走向另一个偏颇的方式。这中间很难权衡采纳的临界点，毕竟游戏是从用户性和市场性出发的。在这里有必要提及游戏的整体出发点是通过开发者预判玩家市场做出的。这是开发者做这款游戏项目的既定思维，而玩家在游戏进程中所遇到的情况包括是否符合玩家的心理预期是另一层面的事情。由此才会带来不同的评价，心满意足的开发者很亮堂地给了高分，而没有获得乐趣的玩家则可能显得垂头丧气。

这当中就需要一个新的权衡点：在不能同时满足所有玩家的情况下是照顾特定群体，还是进行二次妥协照顾绝大部分玩家。但与此同时还有一个顾虑：免费游戏的玩家付费情况集中在为数不多的忠诚玩家身上，不能很好满足他们的绝对需求又将如何？

问题5：玩家的审美视觉和游戏画面风格

游戏进程的娱乐冲击一般由两个层面来替代玩家实现，分别是视觉和听觉。了解和辨析玩家在这两方面的态度将更好地实现玩家意识。

在视觉方面首先涉及美术风格和构图问题，为了使游戏视觉效果达到最佳状态，画

面良好的模式、质感、环境和光照至关重要。只有构图清晰，游戏环境的视觉结构才足以吸引眼球。尽管手机游戏和社交游戏在风格上暂时无法达到传统大型游戏的品质追求，但是并不意味着这种追求不严谨。以Zynga为例，其Ville系列游戏已经使其拥有大眼睛的可爱卡通角色成了公司的标志。一旦形成印记，玩家就会通过图像来识别游戏，比如更改了马里奥系列游戏的色调肯定会让熟悉的玩家产生诸多不适应。

当然，游戏最终呈现的创意机制或者画面效果可能都是基于开发者之间的内部碰撞或与市场的相互妥协。游戏本身是文化载体，但与此同时游戏是一款面向市场的产品和面向用户的服务，很多环节并不能够随心驾驭，而是需要权衡一个承载开发者设想以及市场反响的结果。而这种博弈的最终结果是完成玩家的审美视觉需求。

问题6：界面设计和玩家使用习惯

无论游戏有多华丽的视觉效果，劣质的UI足以将其完全摧毁。开发者设计UI时首先应该考虑的问题是：什么才是对玩家最重要的信息？包括屏幕简单整洁、按键类触控点尺寸合适、为风景和绘制模式提供适宜的资源。首先是尊重玩家游戏感受，包括尊重玩家本身的偏好和已经被其他游戏养成的习惯性动作。

好的游戏内容遇上复杂或不符合用户逻辑的界面设计遭遇用户拒绝。并且可能不止对一款游戏造成影响，甚至可能对开发者发布其他的游戏带来负面影响。Effective UI的调查认为恶劣的应用体验可能损害品牌价值（每款产品都有自身品牌，每个品牌都有下一款应用）。高达69%的受访者认为如果某款应用的产品体验很不尽如人意，就可能对公司的品牌形象产生消极影响，特别是给以后的产品带来不良连锁效应。对开发者而言，每一款产品的质量就显得相当重要。

问题7：用户的社区概念和群体主义思想影响分析

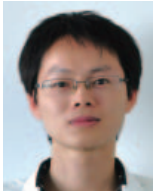
不管是社交游戏还是手机社交游戏都有一个共同的症结：虽然号称Social，但目前各种协作与竞争机制只反映了玩家社群协作性和互动性的冰山一角，并没有办法让玩家之间真正组建一个虚拟游戏社区。特别是依托情感所架构起来的社区。

我们常在说一个道理：一款社交游戏真正的生命力不在于用户是否能自娱自乐获得快乐，而在于玩家是否觉得与更多的朋友分享会更有乐趣。其实我们一直在重复描述的就是群体情感和社区归属感，特别是病毒式传播型的好友邀约方式，将形成一种独特的生态圈。这个生态圈属于小团体交互，因为不得不承认好友间的交互已从现实逐步延伸到虚拟网络，因此就需要思考如何让游戏中的泛友谊更具互动的针对性。

当然游戏群体性的养成可以提高用户黏性和留存率，这两者对用户终身价值具有决定因素。玩家参与度越高，停留时间越长，其潜在终身价值就越大。这种群落关系不一定是结盟形式，也可能可以是密集交互的支撑架构，不管是竞争还是协作，只要玩家在游戏中经常以各种形式接触，其社区依赖性就能产生。

问题8：声效的正面效能和负面效能解析

音效对玩家游戏情感的捕捉具有很大的促进作用，去掉音效，游戏的沉浸性就会缺失。音效具有影响潜意识的功能，即使玩家没有看到游戏画面，特定的音效也能够让他回想起游戏的愉悦感。 游戏的声效同样关系到玩家的进程节奏，玩家的进程节奏可以跟随音效的递进来判定完成度。不仅仅让玩家感受了游戏的氛围，同时也成为了游戏体验的一部分。游戏中的节奏感存在并不在于人为的间隔设置，而在于玩家在游戏体验和游戏进程共鸣中呈现出来的节奏。好的声效便能协助玩家找到这种游戏节奏。



郑金条

游戏邦负责人，游戏邦主要关注和解析国内外社交游戏和手机游戏领域，并定期做深度行业阐述。

责任编辑：陈博（chenbo@csdn.net）

论Android产品高效开发之路

文 / 姚冬

本文通过对Android产业开发的简要分析，总结了Android系统开发中遇到的各种问题，从产品组合、需求管理、建模与重用、协同开发等角度分析和总结出能够加速Android系统开发并有效提高产品质量的方法和思路。最后在此基础上，提出了Android系统开发平台的解决方案，并结合整个产品生命周期，对Android系统开发中的各个环节加以表述。

经过多年的发展，Android已不是最初设想的仅仅作为移动设备的平台，它提供了一个跨设备的通用结构，并且开始进入其他数字消费设备领域，诸如平板电脑、移动互联网设备、Google TV等，甚至有配备触摸屏的家用固定电话、遥控器、饭店的触摸屏目录、医疗设备以及厨房电脑等。

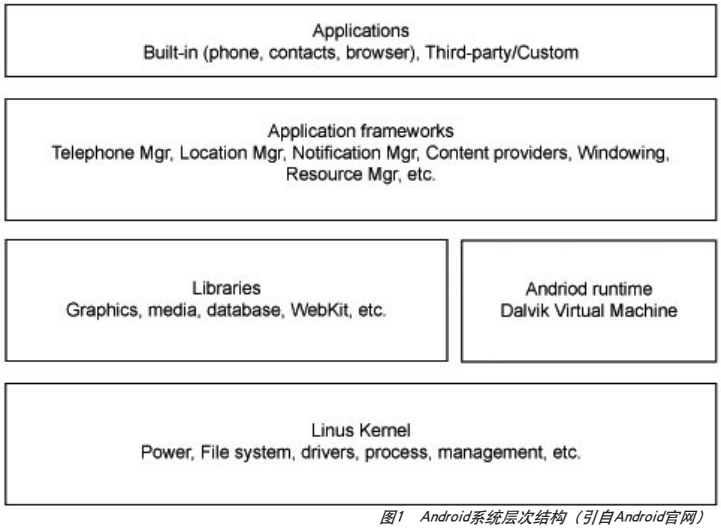


图1　Android系统层次结构（引自Android官网）

Android用它开放、跨平台的通用软件架构征服了众多消费者以及厂商，前景一片光明，但并非一片坦途。无论是与Apple平台，还是Android同一产业链上的不同厂商之间，竞争始终存在。在Android面前人人平等，如何能够在相同的起跑线下脱颖而出，得到市场和客户的认可，是摆在我们面前的巨大挑战。

本文从多个角度出发，分析了Android产品开发中遇到的问题，探讨了有效提高产品开发效率的方式和方法，试图找到一个通用并行之有效的解决方案。由于整个方案贯穿Android产品生命周期，因此称之为Android系统开发平台，希望能够对致力于Android产业的厂商或个人有所启示。

Android系统开发平台

Android希望能够建立标准化、开放式的软件平台和操作系统，平台由操作系统、中间件、用户界面和应用程序组成，已经衍生出一条由手机和其他终端制造商、移动运营商、半导体公司、软件公司等组成的、完整的产业链和生态环境。此举必将推进更好、更快的创新，为移动用户提供更多功能强大的应用和服务。如图1所示。

Android开发不只局限于Android应用的开发，而是涵盖整个Android产业链的所有环节，从系统的定制，到电子设备的研制，甚至可以包含适用于Android平台的各种半导体部件。由于终端和应用最受关注，并且也是最容易被直接接触到的，因此本文会更多地面向设备制造商以及应用开发商。

Android系统在逐渐稳定和成熟，但版本变化还是相对比较快，开发人员需要适应快速变化的版本；同时，Android用户与传统平台的用户群不同，其消费行为也必然不同，要求厂商对目标客户群的消费行为作出良好判断；另外，Android平台上应用开发的盈利模式不够明确，给产业链带来未知因素。Android带来机遇

的同时也带来了挑战，必然不会一帆风顺，对于终端研发以及应用开发会提出种种的挑战和更高的要求。

支持跨平台的应用开发

Android推出至今，历经了多个版本，同时还存在由Android系统衍生出来的特定厂商的系统，这往往要求设备和应用厂商同时支持多个版本。另外对于应用而言，有时不仅需要支持Android，还需要支持其他系统例如iOS、Windows Mobile等。因而提出跨平台的产品和应用的开发，以支持多种平台、多个版本的需要。

产品准确定位，及时响应市场需求

想想近十年Motorola和Nokia等公司的沉浮，让人不禁感叹电子设备市场的瞬息万变。要把握好市场的脉搏，就要求产品准确定位，在对需求加以捕获的同时，进行良好的分析、排优和管理。

产品的创新性要求高，准确捕获与把握创意

电子产品更新换代的速度极快，厂商无法依靠品牌本身来黏合客户，消费者对产品的忠实程度取决于产品质量和功能，这就对产品的创新性提出了更高的要求：要求细致研究客户消费行为，准确寻找市场需求，及时的捕获并把握产品创意。

缩短产品上市时间

电子消费市场日新月异，时间就是生命。要想尽量缩短产品从创意产生到推向市场的时间，就带来了产品的协作开发、功能复用以及测试一体化等需求。

基于上述种种分析，结合Android开发特点，我个人认为有必要建设Android系统开发平台。下文试图从产品组合管理、需求管理、模型驱动开发以及协作开发等方面入手对其进行阐述和分析。限于篇幅和个人研究水平的局限，不求事无巨细面面俱到，只是重点介绍个人心得体会，希望对大家能够有所启示。

通过产品组合管理准确产品定位

变化的市场对所有厂商都提出了挑战，无论是对于终端制造商还是应用开发商而言，如何保持自身的竞争力是面临的最大痛点之一。这就要求企业在不断发掘新创意的同时，提升



图2 Android系统开发平台

生产及开发的效率，建立区别于竞争对手的竞争优势。

对于Android而言，即使是在同一个平台、同一生态环境下的不同厂商之间，也是合作与竞争同在的关系，如何良好地适应竞争激烈的生态环境，就需要依赖对自身产品的组合规划、对产品特性的差分定位、对竞争对手的优势分析。这就引出了产品组合管理的概念。

产品组合管理通过对市场环境的分析，收集商机，捕获并分析需求，根据对竞争对手以及自身企业及产品特性，制定产品组合策略，以期平衡整个产品开发投资组合中的风险和回报，最大限度地提高市场认知和投资收益。

产品组合能够从四个方面创造企业价值：价值最大化、均衡、战略一致性、合理的产品分布。价值最大化允许主动管理组合中的项目，以便从业务目标的角度最大限度地提高组合价值；均衡的产品组合能够防止在决定项目规模时出错，以防出现项目执行时间过长或者项目的业务风险过高等问题；为了实现战略一致性，目标产品组合应以支持公司实现主要业务目标为标准；应确保提供合理数量的资源来支持已经确定的产品项目。

产品特性组合

电子产品非常强调特性，例如是否具备WiFi、有没有拍照功能等。产品的特性讲究人无我有、人有我新。这要求我们一方面加以创新，去产生吸引消费者购买行为的新特性；另一方面也要求我们对产品的种种特性，根据不同的市场定位加以组合，因为并非一个大而全

的产品就可以满足所有人的需要。产品的特性大致可以分成下面三种类型。

- 功能型特性，例如拍照功能、上网功能。
- 性能型特性，例如CPU速率、内存大小、拍照的像素等。
- 合规型特性，例如不同国家的入网要求。

各厂商应对不同类型的特性，根据相应消费人群使用习惯，以及不同市场的合规型要求，进行灵活合理的组合，从而得到适应不同的市场定位的产品组合。

可行性分析

一款产品的推出，除了要具备相应的特性以外，也要对其可行性进行分析。衡量的因素有众多维度，例如技术可行性、商业可行性、市场潜力、财务回报、产品复杂性以及战略一致性。

价值最大化允许主动管理组合中的项目，以便从业务目标的角度最大限度地提高组合价值。

每个企业都有定义的企业战略，对自身及其产品都有相应的定位，产品要求与企业战略目标保持一致，对实现企业战略目标提供帮助和支持。同时产品的投入要求有匹配的产出，成本与收益一定要达到良好的比率，并且要求在收益与风险之间达到平衡。产品技术方面要评估现有技术与预期目标之间的差距、实现的技术复杂度与产品总体架构是否一致。而风险的管控同样是对产品评估的重要因素之一，包括质量风险、进度风险、组织风险、技术风险等。

不同的维度，在不同的市场环境下，面向不同的目标人群，其优先级以及评判标准是不尽相同的，这就需要引入灵活的评判模型，通过产品或特性的两两比对得到打分模型，并针对不同的判决因素进行加权，综合评比得到最终的判决。

资源配置与管理

产品组合管理是联系战略决策和战术执行的纽带，产品开发是商业战略的表现形式，企业实现其战略的重要途径就是通过恰当的展品开发，产品组合管理要求在企业战略与产品开发之间建立桥梁。

企业往往会同时有开发多款产品的需求，而各种资源总是有限的，产品组合管理同时研究资源的分配，帮助企业合理有效的配置资源。资源管理要求能够对产品进行排优，从而优化资源投入与产出，实现产品价值最大化。错误的产品规划将在运行资源、定位、宣传、销售和支持方面给企业带来沉重压力，并行产品组合的增多会加剧运行的复杂性，必须在产品开发项目的数量与公司自身能力之间找到最佳均衡点。基于产品开发能力来调整产品组合规划可以帮助有效提高整个产品开发项目的成效。

通过需求管理和重用缩短产品上市时间

需求的捕获

需求管理首先是对需求和灵感的捕获，面对复杂多变的市场环境，要求准确地捕捉和发掘Android用户的真实需求，捕获业务人员、设计人员的灵感，从而避免因需求的遗漏或是灵感的丢失，错误或模糊的需求所造成产品的延迟或取消。对需求的描述通常通过文字与模型并存的方式进行，文字的优势在于描述具体、精确的事务，例如数字、指标等；而模型的优点在于无歧义、规范化。结合两者优势可以帮助业务人员或产品经理准确地把握和阐述用户需求。

需求的审批

需求需要经过排优、审批的过程，并非所有需求和灵感都切实可行，一个好的想法同时也需要恰当的市场环境。快速变化的市场环境要求简洁灵活的需求批复、实现、追踪流程，保持对市场的最高灵敏度，避免笨重的管理流程带来滞后。另外，不同的需求是针对不同的客户群，其定位和时效性都不同，这就需要通过不同的产品加以实现，这一点至关重要，可以通过前面论述过的产品组合管理得以实现。

需求的复用

在Android的开发过程中，有效的需求复用可以很好地缩短整个软件开发周期。通过需求管理流程，需求得以在不同的产品中实现复用，并

且很好地支持设计开发的复用流程。产品线工程是Android产品系列开发的重要方法，需求复用的管理是支持产品线工程的必要手段。

需求的复用可以针对不同的层面。项目一级需求复用，是将业务规则、数据元素、用户界面等定义为单一的实例且通过需求开发过程为项目成员共享和复用；组织一级的需求复用，是当团队在多个项目协作时，信息可以通过共享的需求库进行重用。这样的需求库可以被组织来为不同产品定义特性。

通过跨平台应用开发减少重复工作量

自推出到现在，Android历经多个版本，前期的系统版本无论是从功能还是稳定性都差强人意，直到1.6版本才有了第一个稳定且功能强大的系统版本。除Android自身这么多版本以外，还存在由Android系统衍生出来的特定厂商的系统，存在很多厂商基于Android某一个稳定版本的基础加以定制，得到的操作系统，例如中移动的OPhone系统、联想的乐OS等。

开发厂商，尤其是应用开发商，有时需要同时支持多个版本的多种系统，多线作战对于开发商而言是极大的挑战。跨平台的应用开发可以帮助开发商良好地应对多平台多版本的应用开发支持，从而提高开发效率，减少重复的开发工作。

跨平台开发的实现需要借助工具的支持，对工具而言要求包括两个内容。

首先，工具需要支持代码生成并通过编译后，可以运行在不同的系统和平台上，并且能够支持跨平台的部署和调试。通过跨平台的编译、部署和调试，极大地帮助开发人员提升开发效率，并且同时能够保障应用开发的质量。

其次，由于不同的系统和平台，甚至Android不同的版本之间，其API调用都有所不同，如果针对每一个版本都需要调整代码，那么将对后续的维护和升级都造成严重的困难。因此，跨平台的开发要求工具能够支持一套设计、一套实现能够应对多种系统和平台，从而减轻重复而枯燥的开发工作。而能够实现这一要求的最佳选择就是模型驱动开发。模型驱动开发工具可以根据不同的目标环境，基于相应的框架和模型，生成适用于特定目标环境的代码。这对于手机开发这样的嵌入式系统开发而言是至关重要的，尤其是当需要适应不同的目标硬件、系统环境时，模型驱动开发由此极大地提高了软件的复用度，使相同的设计和功能模块，可以方便快捷地应用部署到不同的产品设备中。同时由于是基于同一套设计和实现生成的代码，其功能、质量以及稳定性都可以得到良好的保障，从而极大地减轻开发人员的重复工作量，让他们从其中脱身出来，将精力放在更富有创造性和挑战性的设计与开发工作中。

结束语

Android提供的标准化、开放的软件平台和操作系统是史无前例的创举，其商业价值也逐渐凸显，应用软件数量迅速突破了十万大关。一片赞誉中，Android开发的道路却并非总是坦途，尤其是对Android产业链中的不同厂商而言，无论是设备制造商，或是应用开发商，市场竞争始终存在，优胜劣汰是亘古不变的，如何在众多的同行业竞争对手中突破重围，脱颖而出，与每一个厂商切身利益紧密相关。

以上通过对Android系统开发的简单分析，提出Android系统开发平台的概念，从产品组合、需求管理、建模与重用、协同开发等角度分析和总结，提供一些能够加速Android系统开发，并有效提高产品质量的方法和思路。

参考资料

《项目经理到执行官修炼之道——图解组合管理》宁德军，朱华宇，清华出版社 ISBN 978-7-302-23638-2。

《实时设计模式》（美）Bruce Powel Douglass 著，麦中凡，陶伟等译，北京航空航天大学出版社 ISBN 7-81077-421-2。



姚冬

北京邮电大学软件工程在职研究生。现就职于某知名跨国企业，关注复杂系统开发，以及软件过程改进等方面的研究。拥有12年的系统软件开发和软件工程实践经历。

责任编辑：陈博（chenbo@csdn.net）

全景解读 Android UI自动化测试框架TMTS（上）

文 / 丁津津

本文介绍了淘宝Android UI自动化测试框架TMTS（Taobao Mobile Test System）中的Android部分，包括实现原理、创新点、后续计划以及与目前主流框架Robotium、Android Native Driver的比较。

高质量的软件离不开测试，而自动化测试能大幅提高测试效率，面对移动互联网大潮，如何对Android应用进行自动化测试，成为随之而来的问题。

淘宝测试平台自动化工具团队2011年初就着手开发AndroidUI自动化测试框架，当时Google Code上已经有了一个开源的Robotium框架，但对于淘宝客户端来说存在一些缺陷。自己开发框架的好处在于能根据自身需求定制并及时修改一线测试人员反馈的Bug。

必要的技术调研

站在巨人的肩膀上，才能看得更远，要想提高代码的质量和生产率，最重要的途径就是重用好代码。

首先研究下Android SDK中有哪些东西可以使用。

Android已经提供了三种方式进行测试：Monkey、Monkeyrunner和Instrumentation，三种都可以以自动化的方式进行。

顾名思义，Monkey是一个随机压力测试工具，通过命令行启动，对设备发送伪随机事件，进行压力测试。我需要的是可控测试，而Monkey不符合需求，首先否决掉，当然其中发送事件的方式可以借鉴。

Monkeyrunner也是一个工具，使用Python编写，通过这个工具可以编写Python代码，调用Android的API从而达到自动化测试的目的，当时这方面的资料非常少，团队成员对Python都没有接触过，也尽量希望找到使用Java编写脚本框架，能让测试人员在编写脚本的同时提高Java编程能力，有助于提升白盒测试和Bug定位的能力。

最后剩下Instrumentation，首先这是一个Java类，和UI测试的对象Activity类同样位于android.app包下。通过追踪查看源代码，可以发现在应用中启动Activity的内部实现都是通过调用Instrumentation类中的相关方法。下面是在应用开发中经常使用到的一个方法：

```
@Override
public void startActivity(Intent intent) {
    startActivityForResult(intent, -1);
}
```

它是通过调用startActivityForResult(Intent intent, int requestCode)方法来实现：

```
public void startActivityForResult(Intent
intent, int requestCode) {
    …
            Instrumentation.ActivityResult
ar =
                    mInstrumentation.
execStartActivity(
                     this, mMainThread.
getApplicationThread(), mToken, this,
intent, requestCode);
…
}
```

startActivityForResult(Intent intent, int requestCode)方法是调用Instrumentation类的execStartActivity(

Context who, IBinder contextThread, IBinder token, Activity target,Intent intent, int requestCode)方法来实现，当然这个方法加上了@hide注释:

```
/**
* …
* {@hide}
*/
public ActivityResult execStartActivity(
            Context who, IBinder
contextThread, IBinder token, Activity
target,
        Intent intent, int requestCode) {
    …
}
```

Android也特意暴露这个类的一些方法供应用层使用，特别是测试。比如发送模拟事件、控制Activity的生命周期等。

起初团队曾经想过扩展Instrumentation类作为整个框架的基础，为此我和同事先研究了Android系统和应用的启动流程。因为使用Instrumentation需要在Manifest文件中声明<Instrumentation>标签，比如对包名为com.taobao.test的应用进行测试，所以需要添加如下声明：

```
<instrumentation android:targetPackage="com.
taobao.test" android:name="android.test.
InstrumentationTestRunner" />
```

考虑到可能要用扩展后的标签，我开始研究Android系统如何解析Manifest文件。后来证明这是一个错误的决定，直接使用原有的<Instrumentation>标签即可。由于这时已经涉及Android比较底层的内容，考虑项目进度风险，这条方案被否决，我们决定直接使用Instrumentation类进行整个测试框架的设计。后来Android Native Driver就采用了扩展Instrumentation类的方案，现在回想起来不免有些遗憾。

使用Instrumentation类并不意味着需要直接在这个类的基础上进行设计，因为Android SDK在Instrumentation的基础上，结合JUint框架，封装出了InstrumentationTestCase类。

InstrumentationTestCase类有一系列的子类，支持对Activity、ContentProvider、Service这三大组件的测试。UI自动化测试只需对Activity进行测试，通过筛选选定继承其中的ActivityInstrumentationTestCase2来实现UI自动化测试。

在Instrumentation为基础的测试框架下，测试工程与被测应用是运行在一个进程之中，图1是SDK文档对这个框架的描述。

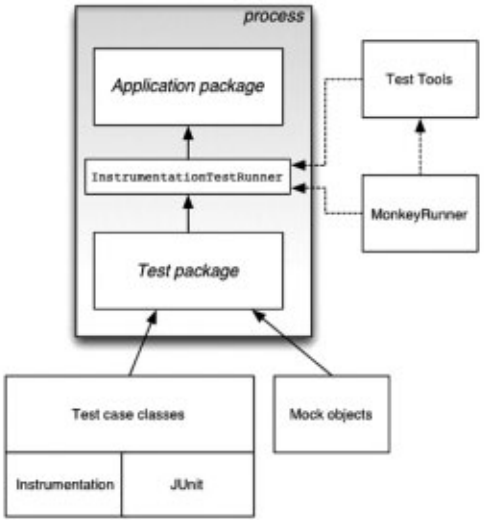


图1 SDK文档对框架的描述

有些应用会重写主Activity的onDestroy()方法：

```
@Override
protected void onDestroy() {
…
android.os.Process.killProcess(android.
os.Process.myPid());
…
}
```

这样会导致一个测试方法运行完后，在ActivityInstrumentationTestCase2类tearDown()时，触发应用Activity的onDestroy()中的这段代码，最终导致整个测试crash。

一开始使用Robotium框架的一线测试工程师就反馈了这个现象，我找出了上述问题的原因，并力图在TMTS框架里面避免这个问题。

要在不更改应用代码的前提下解决这个问题，只能让测试工程与被测应用运行在两个不同的进程之中，进程间的通信问题随之而来。

首先自然想到Android的IPC机制IBinder，后来想到通过HTTP方式实现，但这两种方案又违背了不更改应用代码的初衷。目前这个问题被搁置，通过在测试类中重写tearDown()方法规避。

选定基本的框架后，接下来的工作就是考虑如何进行扩展了。

如何获取Android中的UI元素

之前团队进行Web UI自动化测试，分获得UI元素、操作控件、断言三步。在Android中采

用完全相同的步骤。

Android系统中的UI元素就是android.view.View类及其子类的实例。

很自然就想到在开发应用时使用android.app.Activity类中的findViewById(int id)这个方法。但这个方法需要当前Activity的实例，还要考虑到Activity的跳转，因此这个方案的难点在于实时获取当前Activity实例。幸运的是Instrumentation中的内部类ActivityMonitor提供了getLastActivity()这个API，为实现实时获取当前Activity实例提供了基础。

这里涉及到一个API的易用性问题，想象一下，如果传入的参数是R.java中的整数值，会是一件很麻烦的事，而且数值会随着布局的改变而改变，所谓“己所不欲，勿施于人”，这里传入的id是字符串类型，也就是写在布局文件android:id=”@+id/id”中的id名：

```
<Button
                …
        android:id="@+id/setting"
        …
 >
</Button>
```

然后通过反射拿到其在R.java中的整数值。看来反射不仅在Android应用开发中是一把利器，在测试框架开发中同样是不可或缺的工具。

绝大部分情况下，以上方案是可行的，因为一般需要在测试中操作的控件同样需要在应用中操作，所以应用会在布局文件中为其声明一个ID。但也有特殊情况，比如ScrollView，使用它的目的大多都是为了滚动，并不需要在布局文件中为其声明一个ID；再比如Dialog不需要布局文件就可以创建，对于这些不在布局文件中声明ID的控件，第一种方案就无能为力了。

顺带提一下SDK中自带的hierarchyviewer工具，这个工具能分析当前Activity所加载的布局结构，测试人员可以借助它获得控件的ID。

车到山前必有路，我之前有过Android相关开发经验，首先找到了android.view包下的WindowManager这个类，发现其实是一个接口，顺藤摸瓜找到了其具体实现WindowManagerImpl类，良好的命名习惯确实是能为读懂代码带来很大帮助。

以下是关键的部分：

```
/**
 * …
 * @hide
 */
public class WindowManagerImpl implements
WindowManager {
…
private View[] mViews;
…
     private static WindowManagerImpl
mWindowManager = new WindowManagerImpl();
}
```

这两项正是所需要的，但不仅上述两项是私有成员，WindowManagerImpl整个类也加上了@hide注解。天无绝人之路，Java反射的强大，这个时候又体现出来了。直接通过反射拿到WindowManagerImpl实例，进而拿到View[] mViews，后面发现Robotium中也采用这个方法，乃至最近的Android Native Driver也是采用同样的方法：

```
private static Class<?> windowManager;
static {
  try {
windowManager =
  Class.forName("android.view.
WindowManagerImpl");

  } catch (ClassNotFoundException e)
{
  throw new RuntimeException(e);
  } catch (SecurityException e) {
      e.printStackTrace();
  }
}
public static View[] getWindowDecorViews()
{
…
  try {
  viewsField = windowManager.
getDeclaredField("mViews");
  instanceField = windowManager.getD
eclaredField("mWindowManager");
  viewsField.setAccessible(true);
  instanceField.setAccessible(true);
  Object instance = instanceField.
get(null);
  return (View[]) viewsField.
get(instance);
  }
…
}
```

第二种方案拿到所有控件后，如何找到测试中需要的控件呢？目前只能通过类型过滤返回一个包含指定类型的数组。因为没有在布局中指定ID的控件多数情况在一个Activity中只会有一个，因此通常返回数组第一项是必要的，当然这里提供的API需要传入一个int型参数。

那是不是意味着针对有或没有在布局文件中声明ID属性的不同情况，需要在框架内部使

用不同的实现方式来获取控件呢?

再回过头来看android.app.Activity类的findViewById(int id)方法的底层实现，下面是Activity类中的findViewById(int id)方法：

```
public View findViewById(int id) {
            return getWindow().
findViewById(id);
}
```

它调用了android.view.Window类中的findViewById(int id)方法：

```
public View findViewById(int id) {
            return getDecorView().
findViewById(id);
}
```

可以看到实际上是通过调用android.view.View类中的findViewById(int id)方法来实现：

```
public final View findViewById(int id) {
        if (id < 0) {
            return null;
        }
        return findViewTraversal(id);
}
```

findViewTraversal(int id)方法的内部实现是：

```
protected View findViewTraversal(int id) {
        if (id == mID) {
            return this;
        }
        return null;
}
```

findViewTraversal(int id)在android.view.ViewGroup中被重写：

```
   /**
      * {@hide}
      */
     @Override
      protected View findViewTraversal(int
id) {
        if (id == mID) {
            return this;
        }

        final View[] where = mChildren;
        final int len = mChildrenCount;

        for (int i = 0; i < len; i++) {
            View v = where[i];

                if ((v.mPrivateFlags & IS_
ROOT_NAMESPACE) == 0) {
                 v = v.findViewById(id);

                 if (v != null) {
                     return v;
                 }
            }
        }

        return null;
    }
```

现在来分析一下android.view.Window类中findViewById(int id)调用到getDecorView()的方法，它是定义在android.view.Window类中的的抽象方法：

```
public abstract View getDecorView();
```

这个方法的具体实现在android.view.Window类的唯一子类com.android.internal.policy.impl.PhoneWindow中：

```
@Override
public final View getDecorView() {
    if (mDecor == null) {
        installDecor();
    }
return mDecor;
}
```

下面是和返回值mDecor有关的代码：

```
public class PhoneWindow extends Window
implements MenuBuilder.Callback {
    ...
private DecorView mDecor;
...
private final class DecorView extends
FrameLayout {
...
}
}
```

通过上面对Android源代码分析，可以发现android.app.Activity类的findViewById(int id)方法实际上是通过对一个android.view.ViewGroup子类对象调用其findViewById(int id)方法来实现的。这个android.view.ViewGroup子类对象就是当前Activity所有View的树形结构的根节点。这个过程可以在第二种方案的基础上实现，因此它是一种通用的方案。

在TMTS中，对于在布局文件中声明了id属性的控件，在拿到当前Activity所有控件后使用下面的方法来获取：

```
for (View view : all) {
View result = view.findViewById(id);
     if (null != result) {
         return result;
    }
}
```

上面代码中的ID同样是传入字符类型的ID通过反射到R.java中获得。



丁津津

淘宝软件技术质量部自动化组Android测试开发工程师，从事Android自动化测试框架方面的开发和研究。

责任编辑：陈博（chenbo@csdn.net）



张霞
东软CTO

对新事物要有好奇心

东软CTO张霞专访

记者 / 青耘

1991年，几位教师仅凭3万元经费起家创立东软，现在的成绩有目共睹，其CTO张霞也几乎经历了东软20年的发展变迁。为此本刊对她进行了专访，希望通过她对东软产品技术战略的判断、理解，带大家一窥东软成功背后的秘密。

《程序员》：在这20年的时间里，东软在产品研发技术上经历了哪些重要的转型？

张霞：1996年，我们决定做自己的自主应用软件产品。1997年，开始进入行业应用阶段，到1999年基本完成转型，实现业务向整个行业解决方案的转型。由于每个行业投资都有投资周期，这决定了我们必须同时介入多个行业，此时就要考虑效率问题。于是我们希望能研发出一个使各行业应用都以此为基础的统一技术架构。到2000年，大部分应用软件开始采用两层架构，但产生了很多问题，包括客户端的安装、部署很困难。因为当时国内Windows盗版严重，用户的PC环境不是很干净，导致我们的客户端系统不好用，这让我们开始萌生使客户端软件零安装的念头，为此我们采用了1995年新推出的Java技术做开发。

我们最近在技术架构上的一次重大转折就是云计算。因为我们现在做的不再是单纯的互联网应用，而是物联网应用，其软件架构的要求跟传统的互联网是不同的。对物联网基础架构进行充分的分析，包括关键技术的探索和实现，是我们这几年的研究重点。

在应用领域方面，未来我们也会开始做面向消费者的服务,比如正在设计的类似于手表的运动传感器产品。

《程序员》：你怎样看待自主研发和技术集成之间的关系？

张霞：在过去20年里，我们业务的增长主要靠自主研发，当然不是所有的都从头开始做，我们会集成一些基础设施和平台，但那是围绕东软应用软件的需求去做的。基本上我们研发时主要关注应用软件——东软自己的那部分，其他诸如硬件、存储、平台、数据库、中间件等，会采用集成的方式。

《程序员》：在这20年的历程中，东软是如何进行知识共用的？

张霞：从软件共用的角度，共用的架构必须是标准化的，我们所有的行业解决方案必须遵循统一的架构体系。对于一个应用软件，除了必要的系统设施和软件以外，中间还有一个共性的技术实现，而每个业务线又有共性的业务，需要我们各个行业的事业部去开发。我们正在推动一种软件产品线的方法学，通过它去积累核心业务。要把所谓的实施落地，首先要总结技术架构，包括技术的选择。技术架构的开发不是简单的事，比如对某一个场景的分析，最后可能有多种方案。架构完成以后，就涉及到用什么样的软件工程、沟通流程来构造具体的应用。我们有统一的统计过程，包括做管理产品线、核心资产开发、基于软件产品实施服务、ICDC的解决方案等，都有相关的方法学。公司配置了资深的方法学顾问团队，来负责以上实施。

《程序员》：你认为优秀的CTO需要具备哪些特质？

张霞：优秀的CTO要对任何新事物都特别有好奇心，看到一项新技术时就会马上联想到它未来的用处、可以基于此做哪些创新。作为东软的CTO，可能要求还要更高一些，视野要开阔，能比较早地把一些可能成为下一代的技术做提前储备。

用户体验与敏捷开发的那些事儿

文 / 王亚洲

每个企业和团队应该根据自身情况不断探索、创新，不要拘泥于敏捷开发或者UCD的教条，最适合自己、能够提高开发效率和产品质量的方法就是最好的方法。

人们为什么追求敏捷开发？

敏捷开发是由几个程序员提出的一组软件开发指导原则。它鼓励程序员迈着小而踏实的步子（迭代）接近最终的目标。在迈完每一步的时候，理想情况下都有一个可以运行的软件供客户查看和验证。一个典型的敏捷开发流程看起来像图1那样。

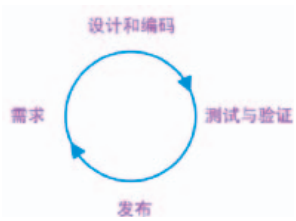


图1　典型的敏捷开发流程

我们可以把敏捷开发想象成一种规避风险的策略。与传统的瀑布开发模型不同，敏捷开发不鼓励前期过多的设计，而是提倡程序员一点点把软件编写出来，在编写的过程中根据实际情况调整项目目标和软件结构，从而可以较好地规避因错误估计可行性、错误估算时间以及需求变更带来的风险。敏捷开发提倡频繁地与客户接触，这样在项目方向偏离之前就可以及时发现并纠正。

人们喜爱敏捷开发主要有以几方面原因。

■ **更快的回报**。敏捷开发的迭代本质决定了在每个迭代周期结束时，都有一个可以运行的软件，这样项目的参与者和客户都可以更快更直观地看到成果。

■ **增强成就感**。程序员认为最重要的东西无非就是实现了某个软件时的成就感。看着一个软件在自己的参与下一点点成长无疑会带来巨大的成就感，而且越到项目后期这种成就感就越强。传统的瀑布模型只有到最后阶段，程序员才能看到可以运行的软件的全貌。

■ **避免软件功能臃肿**。敏捷开发提倡快速，每个迭代中只做优先级最高、最必要的事情，因此可以有效避免因空想和假设而为软件加上许多无用功能的情况。

■ **鼓励每个人的积极参与**。敏捷开发很强调和客户的面对面交流，因此也促成了不同背景的人频繁合作的机会，为创新提供了土壤。另外敏捷开发会逐渐培养一种讲求实际的追求快速解决问题的态度。

以用户为中心的设计（UCD）如何工作？

UCD认为软件系统的需求必须来自它的用户。它有下面两个要点。

■ 在项目定义阶段采用各种方式（焦点小组、现场研究、采访等）进行大量的用户调研，以这些真实的信息和数据来替代我们头脑中现有的对于需求的假设。从开始用户调研到总结出需求往往需要一定时间。

■ 在设计与开发过程中或软件发布之后，

通过正式或非正式的可用性测试，让用户试用软件设计原型或者真实产品，通过观察其行为来发现软件中存在的可用性问题、未被满足的需求或者无用需求，并将此反馈到软件的下一阶段开发中。

UCD的核心也是迭代。系统的设计不是一蹴而就的，而是通过设计、用户验证、修改设计、再验证的循环来不断完善的。从这个方面看，UCD与敏捷开发的思想是完全契合的。

UCD与敏捷开发一个最大的不同：项目初期我们是否清楚的知道需要做什么？

UCD思想中最重要的部分之一就是前期通过用户调研、头脑风暴、市场分析、竞争对手分析等方式，由设计师和产品经理共同确定产品策略和概念设计。这个阶段的成果实际就是项目开发的目标，即“我们要做什么？”而敏捷开发则假设这样的信息在开发开始时已经存在了。

通过敏捷开发和UCD各自产生的背景可以理解这种差异，如图2所示。

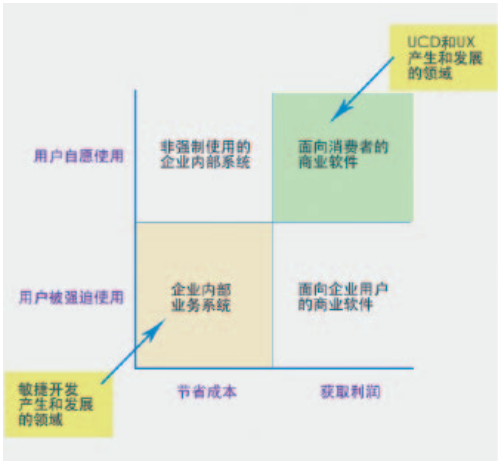


图2　敏捷开发和UCD各自产生的背景
（图表来源：http://www.agileproductdesign.com/blog/emerging_best_agile_ux_practice.html）

通常情况下，软件开发团队对企业内部系统的需求有较多的了解，使用敏捷开发更多的是来改进现有系统，而不是重新开始。开发面向普通消费者的软件，需求了解要难得多。一方面，与本类软件的市场成熟度有关；另一方面，这样的软件系统对创新的要求较高，必须深刻了解用户的痛点，找到真正的问题，才能给出好的解决方案，在竞争中赢得先机。

那么UCD和敏捷开发能很好地协同工作吗？设计师与程序员在敏捷开发的环境下能很好地合作吗？答案是肯定的，只要我们注意下面的一些经验和教训。

值得强调的一点：如果你的企业中本来就没有UCD或者UX的概念，也没有设计文化，那么敏捷开发无法给你带来这些；如果你的团队设计文化或者UCD本来就很强，那么正确地将其应用到敏捷开发过程中也许可以放大UCD和UX的价值。

UCD与敏捷开发如何更好地融合？

每个企业和团队应该根据自身情况不断探索、创新，不要拘泥于敏捷开发或者UCD的教条，最适合自己、能够提高开发效率和产品质量的方法就是最好的方法。

下面给出的是我为值得借鉴的经验。

UX团队的组织架构

总起来说，UX团队有两种组织方式：一是分散的，二是集中的。这两种方式各有自己的优劣。

■ 分散方式，指的是用户体验人员没有一个统一的组织，他们按照项目或者产品线分布在公司的各个角落，接受项目经理或者产品经理的管理。这样做的好处是设计师与程序员的关系亲密，帮助和指导程序员更加方便。例如，设计师看到屏幕上哪个元素的位置不对，可以马上告知程序员，后者可以很方便地修改；程序员对设计文档中某个部分不清楚，可以马上从身旁的设计师那里得到答案。

■ 集中方式，指的是公司内部有一个用户体验或者可用性部门，设计师都隶属于该部门。它更像一个服务性组织，为公司里各个产品或者项目提供服务。这样做的好处是设计师的归属感更强，更加容易管理和在公司内部宣传UX。同时，这使得用户体验部门可以参与公司战略层面的决策，开展公司范围的活动（比如设计指定产品的用户界面标准、建立可用性实验室等），从而使得公司在用户体验方面成熟度得以提高。

在敏捷开发的环境中，分散的方式更有利于协同工作。但我们也不必放弃集中管理方式带来的好处，可以多组织公司内部设计师的聚会、合作，开展跨产品、跨项目的活动等。最理想的情况是设计师和程序员可以在同一地点、同一间办公室。

前期设计

敏捷开发的最大问题就是缺少对软件的整体认识，它在一开始就把工作分成了一个个小块，因此前期设计对于应用敏捷的流程开发软件非常重要。虽然敏捷开发鼓励随机应变，但在迭代开始时如果对要开发的软件没有整体认识，很容易走弯路，甚至开发出无用的产品。前期设计应该在开发开始之前就展开（Sprint 0），利用UCD中的用户调研方法充分理解用户需求，利用Persona、用户故事、用例、工作流模型、用户任务分析等建立起统一的关于要解决的问题（商业需求）的模型。这对设计师、产品经理和开发者理解产品都有很大的好处。一个好的前期设计，至少应该包含上面提到的三项内容，同时，从UX的角度看，前期设计还应包括软件整体的信息架构、灵活而通用的导航设计以及对于通用元素的简略的视觉设计和交互设计。这对设计师来说是个挑战。他必须要可以区分什么是最重要的，必须要在前期设计中完成，什么是可以在后面的迭代开发过程中完成。为了缩短前期设计的时间，对设计师来说，如何为设计任务排列优先级、如何更快地进行用户验证、如何更快地交付自己的设计规约、如何与产品经理和开发经理合作都是尤为重要的。

并行工作

说起来非常简单，UX团队采取和开发团队一样的迭代周期（比如一个月或者两周），但提前开发团队一个周期，如图3所示。



图3　UX团队和开发团队的周期比较

真正的挑战在于，对设计师来说，在Sprint n，他需要设计程序员在Sprint n+1要实现的功能，对程序员在Sprint n中遇到的问题给予支持，同时要审查程序员在Sprint n-1中完成的软件，确保那些没有按照设计实现的功能在以后的Sprint中得到更正。大多数时候这对设计师来说是非常痛苦的。设计师要学会在迭代周期开始之前给自己的工作估算时间，就像程序员做的那样。另外，设计师可以利用开发起来非常复杂的功能为自己争取时间。例如，要实现图形软件中，“另存为其他格式”或者“导出为其他格式”的功能，程序员可能要花费很多时间，但对设计师来说，可能只是一个简单的按钮或者菜单。

轻量化设计

这包含下面几个方面的内容。

■ **拆分自己的工作**。设计师应该学会拆分自己的工作来适应开发团队的迭代周期。在项目初期，可以大致根据功能点或者用户任务来分，后期则更多的是对已实现的功能的调整和完善。

■ **简化设计成果**。如果设计师把设计成果看成是最终要交给程序员的说明手册，势必会包含很详细的设计规约，最终的形式可能是一个很长的、有着插图的文档。这种文档的问题是根本没人会看！设计师可以尝试把设计成果看成是帮助自己理解问题的参考文档。采用低精度的草图、把设计原型本身当成设计规约对与设计师来说足够，而如果设计过程中充分的与程序员进行了沟通，这些东西对于程序员来说也足够了。毕竟在开发过程中，设计师可以随时给予程序员支持。因此，不要把时间花在详细的设计文档上，应提倡快速、有效的交付方式。

■ **尽早开始快速用户验证**。注意这里只能用非常快速的方法，开发开始后，迭代周期中的时间非常宝贵，甚至可以考虑采用RITE（Rapid Iterative Testing and Evaluation）这种被微软、Google和Salesforce.com等很多企业应用的方法。传统的UCD过程非常耗费时间和人力，通常用户研究员和设计师是分开工作的，

用户研究员需要大量的时间来撰写报告并把结果分享给设计师和其他参与者。

■ **设计师在思想上要始终把“轻量级”放在心中**。永远要尝试找出最简单的解决方案。不要为了创新而创造出许多新的概念，因为用户需要时间和精力去学习，如果用户厌烦了，他就不会使用你的产品。Windows有个设计哲学：Reduce concept to improve confidence。其实减少新概念的引入不仅可以增强用户的自信，也可以增强设计师的自信。

教育程序员做设计

通常来说，一个公司或者一个项目中往往缺少设计师。换言之，一个设计师在项目中会承担比较多的任务，往往2~3个开发小组（敏捷开发中的Scrum Team）都由一个设计师来支持。面对如此多的设计任务和支持任务，设计师会应接不暇。一个值得一试的方法是在开发小组中找到有一定设计基础或者对设计有兴趣的程序员并设法得到他们的帮助。设计师可以利用空闲时间对这些程序员进行简单的培训，或者在平时的设计讨论评审中邀请他们参加。这样做的目的是为了在设计师的时间无法调配时，这些程序员知道该怎样做，并指导和帮助其他程序员工作。注意这些培训是不可能让程序员代替设计师的工作的（除非他本身就有天分，而且愿意从事设计师的工作，那样它就是一个设计师），因此不能走向另一个极端，要调整自己对培训效果的期望。另外，为了辅助培训和这些程序员以后开展工作，设计师应该提供界面设计规范甚至一些可复用组件的设计。一个好的例子是，经过培训的程序员可以利用设计师提供的组件设计和界面设计规范设计出一个合格的表单甚至页面。

协同设计

这一点很容易理解但不容易做到。设计师和程序员、设计师和产品经理之间很容易产生冲突，这一切都是因为沟通不够充分。设计师切忌埋头设计，应该采取更开放的方式，多与产品经理和关键程序员讨论。最有效的方式就是给产品经理和关键的程序员看自己的设计草图并详细地给他们讲解，这样一方面可以得到很好的反馈，另一方面，产品经理和程序员对你的设计决定理解得更加深刻，重要的是，他们不会觉得被蒙在鼓里。另外，设计师把自己的工作放在公开的Backlog里让大家看到也是非常重要的。

UX团队的定位

如果你的公司已经把UX放在了一个很重要的位置上，那么在设计你的开发流程时，请体现这一点。最好把UX置于客户或者产品经理那一侧，并明确地说明UX在敏捷开发过程中的职责和话语权。

Minimum Viable Product（MVP）

MVP对任何一个软件开发团队都是一个重要概念。初期在探索软件的方向和愿景时，我们可以无限遐想，自由设计，不受时间、空间、硬件、技术等的约束，但在真正开发时要脚踏实地，甚至稍微变得保守一些。在软件整体愿景明确的情况下，每个版本的目标也要明确，这个目标可以定得小一些，切忌贪多。所谓“Think big, start small”就是说的这个。

总之，敏捷开发与UCD在方法论上有着诸多相似之处，只要我们在应用的过程中注意到设计师和程序员工作方式的差异并加以利用、加强项目参与人员间的沟通、精简设计交付和用户验证的流程，就可以在保证产品开发进度和质量的情况下使产品拥有好的用户体验。



王亚洲

目前就职于SAP中国研究院，从事用户体验、交互设计等方面工作，对Web设计、数据可视化及BI系统设计有所研究。

责任编辑：董世晓（dongsx@csdn.net）

编程技术面试的五大要点

文 / 何海涛

在作者看来，扎实的基础知识、高质量的代码、清晰的思路、优化代码的能力、优秀的综合能力是编程技术面试的五大要点。

找工作一直是一个热门话题。要想找到心仪的工作，难免需要经过多轮面试。编程面试是程序员面试过程中最为重要的一个环节。如果能在编程面试的环节充分展示自己的能力，那么拿到中意的Offer就是水到渠成的事情。

我先后在欧特克、微软和思科等公司任软件工程师，多次接受他人的面试，同时也面试过很多人。总结面试与被面试的经验，我发现尽管面试官的背景、性格各不相同，但都关注应聘者五种素质：扎实的基础知识；能写高质量的代码；分析问题时思路清晰；能优化时间效率和空间效率；具备包括学习能力、沟通能力、发散思维能力等在内的综合能力。

扎实的基础知识

扎实的基本功是成为优秀程序员的前提条件，因此面试官首要关注应聘者的素质即是否具备扎实的基础。通常基本功在编程面试环节体现在两个方面：一是编程语言，二是数据结构和算法。

每个程序员至少要熟练掌握1~2门编程语言。面试官从应聘者在面试过程中写的代码以及跟进的提问中，能看出他编程语言掌握的熟练程度。以大部分公司面试要求的C++为例，如果函数需要传入一个指针，面试官可能会问是否需要为该指针加上const，把const加在指针不同的位置有什么区别；如果写的函数需要传入的参数是一个复杂类型的实例，面试官可能会问传入值参数或者引用参数有什么区别，什么时候需要为传入的引用参数加上const。

数据结构通常是编程面试过程中考查的重点。在参加面试之前，应聘者需要熟练掌握链表、树、栈、队列以及哈希表等数据结构以及它们的操作。如果我们留心各大公司的面试题，就会发现链表和二叉树相关的问题是很多面试官喜欢问的问题。这方面的问题看似简单，但真正掌握也很不容易，特别适合在短短几十分钟的面试时间内检验应聘者的基本功。如果应聘者事先对链表的插入和删除结点了如指掌，对二叉树的各种遍历方法的循环和递归写法都烂熟于胸，那么真正到了面试时也就游刃有余了。

大部分公司对算法的要求都只是考查查找和排序。应聘者可以在了解各种查找和排序算法的基础上，重点掌握二分查找、归并排序和快速排序，因为很多面试题都只是这些算法的变体而已。比如把排序好的数组的前面若干个数字移到数组的后面，然后问怎样在这个数组之中找到最小的数字。这道题其本质就是考查二分查找。少数对算法很重视的公司比如谷歌或者百度，还会要求应聘者熟练掌握动态规划和贪婪算法。如果对这种类型的公司感兴趣，那么应聘者在参加面试之前就应该加强对相关算法题目的练习。

高质量的代码

只有注重质量的程序员，才能写出鲁棒稳定的大型软件。在面试过程中，面试官总会格外关

注边界条件、特殊输入等看似细枝末节但实质至关重要的地方，以此来分析应聘者是否注重代码质量。很多时候，面试官发现应聘者写出来的代码只能完成最基本的功能，一旦输入特殊的边界条件参数就会错误百出甚至程序崩溃。

举个很多应聘者都被问过的一个问题：写一个函数，把字符串转化成整数。这道题看似很简单，绝大部分计算机专业的毕业生都能用十行以内的代码实现最基本的功能。可是在实际面试过程中，十个应聘者中只有一个人能通过这道题的面试，因为绝大部分应聘者不能全面考虑到各种特殊输入，比如输入的字符串含中有非数字的符号、在字符串的开头有正负号、字符串中有正负号但其位置不是在字符串的开头。除此之外，面试官还希望应聘者能考虑的边界条件包括2147483647（0x7FFFFFFF，int能表示的最大正整数）和-2147483648（0x80000000，int能表示的最小负整数）。

除了边界条件和特殊输入考虑不足之外，面试官还有一个不能容忍的错误就是程序崩溃。面试时很多应聘者都会忘记对空指针做特殊处理而导致程序崩溃。如果面试时遇到链表、二叉树相关的题目，应聘者一定要特别小心。因为这两种题目对应的代码里通常会有大量的指针操作，如果考虑不周到，就有可能对空指针进行操作而使程序崩溃。

比如这样一道题：输入一个链表的头指针和一个无符号整数k，输出该链表的倒数第k个结点。这个题目很多人都能想到用两个指针来解决：第一个指针先在链表上移动k-1步，同时让第一个指针和第二个指针在链表上移动。当第一个指针移动到尾指针时，第二个指针指向的就是倒数第k个结点。然而不是每个应聘者都能根据正确思路写出完整的代码。不少应聘者会忽略两种可能：一是输入的链表头指针有可能是空指针；二是链表上结点的数目有可能少于k个。忽略这两点的代码都存在崩溃的可能，从而很难获得面试官的青睐。

要想写出鲁棒的高质量代码，需要在动手写代码之前想好测试用例。在写代码之前，先要想好各种边界条件和特殊输入作为测试用例。当代码写好之后，自己在心里用之前想好的测试用例来检验自己写出的代码，这样就能在面试官之前发现并解决问题。以求链表的倒数第k个结点为例，如果事先想到了输入头指针为空指针和链表上的结点总数少于k这两个测试用例，并且在写好代码之后在心里模拟代码的运行过程，确保能够通过这两个测试用例的测试，那么这轮面试必然是能够通过的。

清晰的思路

只有思路清晰，应聘者才有可能在面试过程中解决复杂的问题。有时面试官会有意出一些比较复杂的问题，以考查能否在短时间内形成清晰的思路并解决问题。对于确实很复杂的问题，面试官甚至不期待应聘者能在面试不到一个小时的时间里给出完整的答案，他更看重的可能还是应聘者是否有清晰的思路。面试官通常不会喜欢应聘者在没有形成清晰思路之前就草率地开始写代码，结果写出来的代码容易逻辑混乱、错误百出。

随着软件系统的规模越来越大，软件开发已经告别了单打独斗的年代，程序员与他人的沟通变得越来越重要。

应聘者可以用几个简单的方法帮助自己形成清晰的思路。

首先是举几个简单的具体例子让自己理解问题。当一眼看不出问题中隐藏的规律时，可以试着用1~2个具体的例子模拟操作的过程，这样说不定就能通过具体的例子找到抽象的规律。

其次可以试着用图形表示抽象的数据结构。像分析与链表、二叉树相关的题目时，可以画出它们的结构图来简化题目。

最后可以试着把复杂的问题分解成若干个简单的子问题，再一一解决。很多基于递归的思路，包括分治法和动态规划法，都是把复杂的问题分解成一个或者多个简单的子问题。

比如把二叉搜索树转化排序的双向链表这个问题就很复杂。碰到这个问题，不妨先画出1~2个具体的二叉搜索树及其对应的排序双向链

表，直观地感受二叉搜索树和排序的双向链表有哪些联系。如果一下子找不出转换的规律，可以把整个二叉树看出三部分：根结点、左子树和右子树。当递归地把转换左右子树这两个子问题解决之后，再把转换左右子树得到的链表和根结点链接起来，整个问题也就解决了。

优化代码的能力

优秀的程序员对时间和空间的消耗锱铢必较，他们很有激情不断优化自己的代码。当面试官出的题目有多种解法时，通常他会期待应聘者最终能够找到最优解。这就要求应聘者在面试官提示还有更好的解法时，不能放弃思考，而应该努力寻找在时间消耗或者空间消耗上可以优化的地方。

要想优化时间或者空间效率，首先要知道如何分析效率。即使是同一个算法，用不同方法实现的效率可能也会大不相同，要能够分析出算法及其代码实现的效率。例如求斐波那契数列，很多人喜欢用递归公式f(n)=f(n-1)+f(n-2)求解。如果分析它的递归调用树，就会发现有大量的计算是重复的，时间效率是以n的指数增加。但如果先求f(1)、f(2)，再根据f(1)和f(2)求出f(3)，接下来根据f(2)、f(3)求出f(4)，并以此类推用一个循环求出f(n)，这种计算方法的时间效率就只有O(n)，比前面递归的方法要好很多。

要想优化代码的效率，还要熟知各种数据结构的优缺点，并能选择合适的数据结构解决问题。我们在数组中根据下标可以用O(1)完成查找。数组的这个特征可以用来实现简单的哈希表解决很多面试题，比如在字符串中找到第一个只出现一次的字符。再比如为了找出n个数字中最小的k个数，需要一个数据容器来存储k个数字。在这个数据容器中，我们希望能够快速地找到最大值并且能快速地替换其中的数字。经过权衡，我们发现二叉树比如最大堆或者红黑树都是实现这个数据容器的理想选择。

要想优化代码的效率，也要熟练掌握常用的算法。面试中最常用的算法是查找和排序。如果从头到尾顺序扫描一个数组，需要O(n)时间才能完成查找操作。但如果数组是排序的，应用二分查找算法就能把时间复杂度降低到O(logn)。排序算法除了能够给数组排序之外，还能用来解决其他问题。比如快速排序算法中的Partition函数能够用来在n个数里查找第k大的数字，从而可以用O(n)的时间在数组中找到出现次数超过数组长度一半的数字。如果面试题是一个求最大值或者最小值的题目，则可以尝试用动态规划法或者贪婪算法，比如用动态规划法求出数组中连续子数组的最大和。

优秀的综合能力

在面试过程中，应聘者除了展示自己的编程能力和技术功底之外，还需要展示自己的软技能，诸如沟通能力和学习能力。随着软件系统的规模越来越大，软件开发已经告别了单打独斗的年代，程序员与他人的沟通变得越来越重要。在面试过程中，面试官会观察应聘者在介绍项目经验或者算法思路时是否观点明确、逻辑清晰，并以此判断他沟通能力的强弱。另外，面试官也会从应聘者说话的神态和语气来判断他是否有团队合作的意识。通常面试官不会喜欢高傲或者轻视合作者的人。

IT行业知识更新很快，因此程序员只有具备很好的学习能力才能跟上知识更替的步伐。通常面试官有两种办法考查应聘者的学习能力。第一种方法是询问应聘者最近在看什么书、从中学到了哪些新技术。面试官可以用这个问题了解应聘者的学习愿望和学习能力。第二种方法是抛出一个新概念，接下来他会观察应聘者能不能在较短时间内理解这个新概念并解决相关的问题。比如面试官要求应聘者计算第1500个丑数。很多人都没有听说过丑数这个概念。这时面试官就会观察应聘者面对丑数这个新概念，能不能经过提问、思考、再提问的过程，最终找出丑数的规律从而找到解决方案。

知识迁移能力是一种特殊的学习能力。如果我们能够把已经掌握的知识迁移到其他领域，那么学习新技术或者解决新问题就会变得容易。面试官经常会先问一个简单的问题，再问一个很复杂但和前面的简单问题相关的问题。这时面试官期待应聘者能够从简单问题中得到启示，从而找

到解决复杂问题的窍门。比如面试官先要求应聘者写一个函数求斐波那契数列，再问一个青蛙跳台阶的问题：一只青蛙一次可以跳上1级台阶，也可以跳上2级台阶，请问这只青蛙跳上n级的台阶总共有多少种跳法？应聘者如果具有较强的知识迁移能力，就能分析出青蛙跳台阶问题实质上只是斐波那契数列的一个应用。

还有不少面试官喜欢考查应聘者的抽象建模能力和发散思维能力。面试官从日常生活中提炼出问题，比如如何判断5张扑克牌是不是顺子，考查应聘者能不能把问题抽象出来用合理的数据结构表示，并找到其中的规律解决这个问题。面试官也可以限制应聘者不得使用常规方法，这要求应聘者具备创新精神，能够打开思路从多角度去分析、解决问题。比如面试官要求应聘者不用加减乘除四则运算实现两个整数的加法。此时面试官期待应聘者能够打开思路，用位运算实现整数的加法。

小结

我们可以用下图来总结出应聘者需要具备的素质。



从上图可以看出，应聘者在面试之前需要做足准备，对编程语言、数据结构和算法等基础知识有全面的了解。面试时如果碰到简单的问题应聘者一定要注重细节写出完整、鲁棒的代码。如果碰到复杂的问题应聘者可以通过画图、举具体例子分析和分解复杂问题等方法先理清思路再动手编程。除此之外，应聘者还应该不断优化时间效率和空间效率，力求找到最优的解法。在面试过程中，应聘者还应该主动提问弄清楚题目的要求，表现自己的沟通能力。当面试官前后问的两个问题有相关性时，尽量把解决前面问题的思路迁移到后面的问题中去，展示自己良好的学习能力。如果能做到这么几点，那么应聘者顺利通过面试获得心仪的职位将是瓜熟蒂落的事情。



何海涛

思科高级软件工程师，之前先后任职于Autodesk和微软。主要关注C++/C#的开发技术，并对设计模式和项目管理也很感兴趣。

责任编辑：董世晓（dongsx@csdn.net）

寻找机遇 创造未来
CSDN 热门职位全新推荐

UFIDA 用友

用友软件

诚聘以下职位：

一、Java开发工程师

岗位要求：

1. 两年以上Java开发经验；
2. 熟悉SQL Server或Oracle数据库；
3. 熟悉Eclipse开发工具；
4. 熟练掌握SQL语句，能够熟练进行数据库设计。

二、.NET开发工程师

岗位要求：

1. 两年以上.NET开发经验，具备一定分析设计能力；
2. 良好的C#代码实现能力，熟悉ASP.NET开发技术；
3. 掌握面向对象的设计方法和设计原则。

简历投递：rdzhaopin1@ufida.com.cn

DeNA China

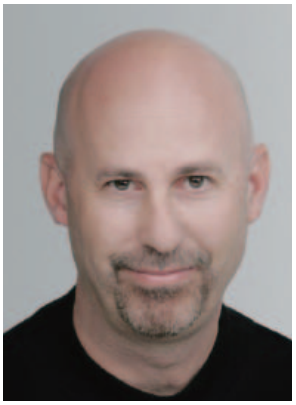
上海纵游网络技术有限公司

我们正在全力招聘中国人才，打造一个全新的中国团队，致力于中国智能手机游戏平台的构筑和运营。愿意加入我们，一起构筑世界著名的智能手机游戏平台吗?

Android/iPhone/Web application，更多职位招聘信息请登录公司主页进行查询

公司主页：www.denachina.com

请以“职位+姓名+CSDN”作为应聘邮件标题，将中英文或中日文简历发至
hr@denachina.com



网络产品的内涵与产品机会评估

文 / Marty Cagan 译 / 姜沈励，周蓉，魏琼

Marty Cagan
过去20年，Marty Cagan作为负责定义和开发产品的高级经理人为多家一流企业工作过，包括惠普、网景通信、美国在线、eBay。他亲历了个人电脑、互联网、电子商务的起落沉浮，致力于通过写作、演讲、培训帮助客户打造富有创意的产品。为此，他撰写了《Inspired: How to Create Products Customers Love》一书，创办了硅谷产品集团公司（SVPG）。在此之前，他的最后一份工作是担任eBay产品管理及设计高级副总裁，负责规划全球电子商务网站的产品和服务。

Marty Cagan是享有世界声誉的产品管理专家，曾经担任网景副总裁、eBay产品管理及设计高级副总裁。本文是他回顾自己二十多年来从事软件产品管理工作的总结和经验分享，谈到了他对产品的理解，以及评估、挑选产品的经验。

产品，你理解对了吗？

在大众消费品行业，提到“产品”一词时，大部分人脑中会浮现出一个非常清晰的实体，例如一块肥皂或是一把剃须刀。但在互联网行业，“产品”一词就显得格外抽象：首先，互联网上的产品通常是指软件，而不是实物；其次，这些软件通常并不安装在你的电脑里，而是直接运行在远端服务器上，甚至是运行在抽象的“云”里。

大部分网络公司由于网站体型过于庞大，需要人为地将其拆分成许多小“产品”。大型网络公司的网站一般被拆分成许多（通常是几十个）小“产品”，这种组成结构只是为了适应内部组织需求（Organizational Needs），而普通用户是察觉不到的。

在以前的文章中我提到过把大型网站拆分成多个可管理模块的关键点。现在，我想谈谈对产品的看法：当我们提到“产品”时，到底指的是什么？ 这里的产品可以指整个网站也可以指其中一个模块。

我觉得今天讨论的对象相当重要。我认为，大部分团队只是凭直觉工作。他们定义产品和组织团队的方式不合理，直接造成了好产品“难产”。在采访过这类团队后，我觉得，让大家意识到什么是产品，也许能改善这种现状。

我这样定义典型的网络公司产品：

产品 = 用户的整体体验 = 功能 + 设计 + 盈利 + 内容

- 功能是指产品可以干什么。
- 设计是指交互设计、视觉设计和工业设计（对物理设备而言）。
- 盈利是指盈利模式。可能是各种形式的广告、捐赠或者交易。
- 内容可以是自己原创的、由用户提供的，或者两者相结合。

虽然看似简单，没有火箭制造技术那么高深神秘，但对我来说，还是存在两个关键点：

- 产品的外观和功能不可分割，它们应该被视为一个整体；
- 网站的产品负责人必须平衡好功能、设计、内容与盈利之间的关系。

比如说：

- 如果设计师不是和产品经理、工程师密切合作，那么每个人的工作都不好开展；
- 如果强行将网站员工划分为两个团队，一个团队负责开发为用户提供价值的产

品，另一个团队负责营销推广，你会发现很难优化整体的用户体验，甚至会引起两方的利益冲突；

■ 如果只专注于功能开发，而忽略了内容的创造，你就无缘感受整合带来的好处。

此外，请注意：

■ 在电子商务中，对“产品”的定义除了上述四点外，还包括订单履行（Fulfillment）、商品包装、客户服务、退换货策略，并且，我们常常把网站上出售的货物叫做“商品”，避免和这里所提到的“产品”相混淆；

■ 在传媒公司，编辑就是内容的缔造者，而内容是产品的一个组成部分；

■ 与此同时，公司还需要有专业化的产品团队（具体内容见后续的文章），并且团队的“性能”通过产品积分卡（Product Scorecard，具体内容见后续的文章）来衡量。

作为互联网行业的产品经理、设计师或者开发者，如果你能用系统的眼光来看待产品功能、设计、盈利和内容，必将受益匪浅。

评估产品机会

市场给予新产品诸多机会，即使成熟的市场也不例外。因为市场环境充满变数：竞争对手不断被淘汰，新技术、新创意不断涌现……

产品经理必须用他灵敏的“嗅觉”，从纷至沓来的机遇中迅速评估、挑选出有市场潜力、可行的创意，过滤那些没有价值或时机尚不成熟的点子。

大多数公司的产品选择权掌握在高管手中，这类决定不容置疑，只能执行；有些公司的产品选择权由市场部门掌握；还有些公司的产品创意来自开发团队。在这些情况下，评估产品机会的流程往往被忽略，一切全凭直觉判断（更有甚者，只要大客户提出要求，项目就直接上马）。

正常情况下，业务人员会撰写一份论证产品可行性的市场需求文档，描述待解决的问题。理论上，市场需求文档只描述产品机会，不涉及具体解决方案。但多数公司跳过了市场需求文档，有些将市场需求文档误写成产品规范文档，有些则没回答该回答的问题。即便完成了市场需求文档的公司，也常常将它束之高阁，不闻不问。

结果出现大批绕过市场需求文档，回避评估产品机会，直扑产品经理。凡事须三思而后行。想快捷、高效地开发产品谈何容易？

评估产品机会是产品经理的重要职责。评估产品机会的目的在于：淘汰馊主意，避免浪费时间和金钱；挑选合适的产品机会，团结团队，理解产品，整合资源。

作为互联网行业的产品经理、设计师或者开发者，如果你能用系统的眼光来看待产品功能、设计、盈利和内容，必将受益匪浅。

为了评估产品机会，我要求产品经理回答以下十个问题。

■ 产品要解决什么问题？（产品价值）

■ 为谁解决这个问题？（目标市场）

■ 成功的机会有多大？（市场规模）

■ 怎样判断产品成功与否？（度量指标或收益指标）

■ 有哪些同类产品？（竞争格局）

■ 为什么我们最适合做这个产品？（竞争优势）

■ 时机合适吗？（市场时机）

■ 如何把产品推向市场？（营销组合策略）

■ 成功的必要条件是什么？（解决方案要满足的条件）

■ 根据以上问题，给出评估结论。（继续或放弃）

以上问题并不涉及具体的解决方案。机会评估只讨论待解决的问题，不应涉及具体解决方案。将来有大把的时间来考虑解决方案，现在是认真考虑要解决什么问题的时候。产品经理往往把待解决的问题和解决方案放在一起考虑，当具体解决方案遇到困难时，他们会放弃产品机会。这是典型的“把洗澡水和孩子一起泼掉”的做法。

最难回答的往往是机会评估的第一个问题

——产品价值。很多人感到惊诧，这应该是最容易回答的问题呀！问问产品经理产品要解决什么问题，你会发现多数人答非所问，只能泛泛地说出产品的功能和特色。

另一个棘手问题是如何评估市场规模。你可以求助于行业分析师、贸易协会、公司财务，再结合自己的分析作出判断。评估务必谨慎，避免浮夸，不是所有产品都有十亿美元的市场。

营销组合策略也很重要。它描述具体销售方式，甚至会影响产品需求。

成功的必要条件（解决方案要满足的条件）指的是在调研过程中发现的特殊需求。同样，确定必要条件的任务不是描述解决方案，而是搞清楚产品的依赖因素和约束条件。比方说，如果要通过系统集成商来销售产品，对方可能会对产品的扩展性、合作方式提出要求。

说白了，产品公司的任务就是挑选合适的产品机会，然后向用户提供实用的解决方案。产品经理负责评估各种创意，在公司投入宝贵的时间和金钱之前，淘汰蹩脚的创意，找出对公司最有利的机会。

产品的机会评估结果出来后，别忘了呈报给公司高管，与高管讨论，决定是否开发此产品。如果决定继续开发，了解高管的想法，有助于你进一步开展工作。

如果CEO告诉你不管愿不愿意，都要继续开发，该怎么办？在这种情况下，迅速进行机会评估，明确产品需求也是必要的。你得到的结论可能会改变CEO的看法，即使不能，至少你能明确产品目标，大大提高产品成功的可能性。

开发新产品还是维护旧产品？

经常有人问我在开发新产品和改善原有产品之间如何取得平衡。他们希望知道具体比例。我认为应该换个角度来考虑这个问题。对我来说，所有的项目，不管是开发新产品，还是改善原有产品，都属于产品机会，与新旧无关。我们要考虑的是哪个机会更好。

开发新产品能为老用户提供更多选择，还能吸纳新用户；改善原有产品能提高老用户的满意度，也能吸纳新用户。两者各有千秋。

关键在于比较两者的机会。产品团队一视同仁地评估两者的收益与成本，然后由管理团队作出决策。新公司多半会选择新的产品机会，具有一定规模的公司多半会选择完善现有产品。在选择产品机会时，机会主义并不是不可取的。

很多时候，好机会就在眼皮底下，完善不尽如人意的产品特性往往事半功倍。举例来说，100位打算注册使用产品的用户最后只有9人顺利完成注册，如果设法把人数提高到18人，就能让产品的收益翻倍。这种改进往往很容易实现，只要做一些原型测试和用户测试，很快就能找出存在的问题，想出解决方案。

再举一个例子，产品公司往往雇用上百人的客服团队为用户提供售后服务，如果能提高产品的易用性，将大幅减少客服人员的数量，降低成本，同时提高用户满意度和用户净推荐值。

每当我指出这类“机会”、提高公司利润后，都会被看做英雄。这主要是因为软件公司过于自负，以为产品已经足够好，继续投入也不会有大的改进。他们要么认为产品非常复杂，根本无法改进；要么满足于9%的注册成功率，宁愿花钱做营销，投放广告；要么认为客户服务的支出是必不可少的。实际情况往往是产品缺乏竞争力，公司只能在其他方面寻找补救措施。

从另一个角度看，这是糟糕的产品设计和蹩脚的用户体验导致的结果。更通俗地讲，就是原有产品质量差，公司不愿意想办法改进，反而认为开发新产品更容易，导致原有产品无法发挥潜力、产生应有的利润。除非这些公司改变研发产品的方式，否则，新产品难免重蹈覆辙。

产品规划七宗罪

为了产品规划，每个产品公司都没少操心。它涵盖了一系列活动：商业战略、产品战略、产品路线图、产品组合管理、机会评估、

项目规划和跟踪，以及项目监督。

但说白了，产品规划就是决定要投资哪些项目。

我正在写的一系列关于产品规划不同层面的文章，但这里我想从产品公司最常遇到的一个问题开始。

不论采用什么产品规划流程，挑出好创意、摒弃烂点子都是你的首要任务。在前面我们已经讨论了产品探索，强调产品经理的首要职责就是发现有价值的、有用的、可行的产品。如果你没有找到了这样一个产品，其他一切努力都是“无用功”。

显而易见，这就意味着，那些不被看好的项目应该在实施之前就被强制“流产”。

但有些应该被“流产”的项目却没有被终止，让公司蒙受了巨大的损失。

我总结了一下，原因大致有7个。

■ 惰性——终止项目或烂点子比起让它继续，遭遇的阻力更大。决定一个项目的去留，需要充分的理由和翔实的数据来作支撑，但很少有团队做到这点。

■ 刚愎自用——很多人不喜欢反对意见，也不肯面对客户的批评。面对批评，他们总会自我安慰，并且一厢情愿地认为一旦产品发布了，用户肯定会喜欢。

■ 爱面子——很多人认为项目终止意味着失败，而失败是可耻的。但我认为让注定会失败的项目继续进行，浪费时间、金钱在毫无价值的产品上，亲眼看着它在市场上被对手绞杀干净，那才叫做“一败涂地”。如果你没能发现好的解决方案，或者用户的反应和你之前的预想有差距，立刻终止项目是明智的举措。毕竟，适时止损比一败涂地要强太多。

■ 尸位素餐——很多产品负责人把管理者的职位当成了太师椅，在其位却不谋其政。这些产品负责人为了坐稳自己的位置，很少冒险做出那样“不得人心”的决定，从而放弃了他们作为产品负责人应该承担的责任。

■ 企业文化所致——一些企业认为终止“鸡肋”项目不符合自己的企业文化，何况已经为这个项目投入了这么多资源，付出了这么多努力。但好的企业文化应该引导员工鄙视这样的行为：在注定要失败的项目里浪费最宝贵的资源和人力。把人员投入到这些项目中，是对员工的不公或者不尊重。

■ 计划不周——“下周工程师就要完成工作了，总不能让他们闲着吧！”这无疑是“铤而走险”的自杀式行径，荒谬却很常见。这时，制订一个有意义的待完成的工作计划（Backlog）显得尤为重要。如果有一个团队空闲下来，而你的项目还没有准备好，那么你可以安排他们去做其他有价值的工作。

■ 盲从——我们要面对这样的现实，很多项目像是老板的宠儿，他自以为这个项目很重要。不要以为他们是老板，就肯定是对的。虽然很多伟大的产品来自于聪明的老板，但“人非圣贤，孰能无过”。

产品探索和创新就像种地。你希望精心培育的点子能够丰收，就必须少播种那些“杂草般”的烂点子，防止它们从好点子那里抢走“营养”。

目前，很多公司同时运行着太多的项目，有好的也有烂的，烂点子无疑会消耗公司的资源，以至无法集中力量实施好的点子。

归根结底，公司决策者要履行职责，终止糟糕的项目，把资源都投入到看得到希望的产品上。

投入到烂点子上的时间和金钱是显而易见的，但很多人没有意识到更大的损失是优势项目的机会成本——你本可以花那些时间和金钱做出更有价值的产品。可是，点子时好时坏，很多团队在初期都难以辨别，只有实施后才知道。

想要改变这样的窘境，产品探索是个不错的方法。而管理团队则需确保开发、测试等宝贵的资源都用在了刀刃上。



本文节选自华中科技大学出版社《启示录：打造用户喜爱的产品》一书和作者的博客。该书从人员、流程、产品三个角度介绍了现代软件（互联网）产品管理的实践经验和理念。特此感谢华中科技大学出版社与Marty Cagan先生授权。

产品管理的前世今生——今天

文 / 张智渊

在《产品管理的前世今生——昨天》一文中，介绍了PMS产生的市场原因、理论依据以及所属的学科范畴。在本篇中，将重点介绍目前国内PMS的发展现状，主要是从企业和从业者两个角度进行说明。建议所有产品管理者、采用或者即将采用PMS的企业阅读。

国内PMS的特点

PMS是随着工业化生产由“生产”为中心转变为“消费者”为中心而产生的，对企业来说，根本目的就是在资源有限的情况下生产出“适销对路”的产品，并保持长久的盈利和发展。

而要实现这个目的，经过市场的选择，PMS就成为目前最适合于企业发展的组织结构。从宝洁1927年提出PMS以来，经过了80年的发展，PMS随着市场情况的不断变化而逐步丰满和完善，并且为全球各行各业所认可和接受。

具体到国内，通常的观点是PMS是从20世纪90年代末进入中国，首先由医药行业兴起【注：该观点是国内产品管理界普遍的认识，具体不作考证，只作参考】，但由于医药行业的特殊性，并没有在全行业推行开来。而随着国外各种管理思想的渗透和新兴行业【注：特指互联网、软件、硬件和通信等行业】的发展，PMS开始在国内逐步发展起来，具体表现为以下特点。

行业发展不平衡

在国内，普遍的认识是“医药、消费品行业是PMS的国内首发行业”，直到现在，这些行业也是PMS的忠实执行者。除此之外，新兴行业，尤其是和国外企业接触比较多、差距比较小的行业，例如IT、通信、金融、家电等行业，都已经普遍采用了PMS，并且显现出愈演愈烈之势。但国内一些传统行业，则表现不太明显，如图1所示。

地区发展不平衡

在国内，经济发达地区的企业采用PMS的数量要远多于经济相对落后地区的企业，CPRC做过一个统计，对其目前的会员进行抽样调查，发现北京、上海、广东三地的产品经理数量占到了调查数量的60%以上。

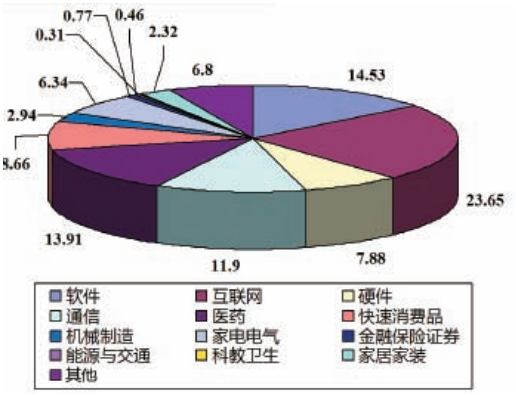


图1　国内产品管理者所在行业分布

企业发展不平衡

即使是同一个行业内的采用了PMS的不同企业，其认知度和应用能力也是不一样的。而这种认识和应用水平上的差异，直接导致了PMS在这个行业中的影响度和扩展性。

PMS发展比较好的企业都能够优先提供资源进行保证，而在PMS发展比较差的企业中，PMS部门已经沦落为“鸡肋”的角色，有些企业甚至开始怀疑PMS是否能真正为企业带来价值。

模仿照搬严重

PMS进入中国时间并不长，而被中国企业所采用，无非就两个原因：第一是跨国企业进入中国，设立PMS是为了保证全球体系一致；第二是中国企业希望学习国外企业的先进管理思想，主动引进PMS。

但无论是哪一个原因，都必须有一个前提条件，就是一定要“实事求是”，不能“照搬模仿”，否则只会出现“水土不服”。

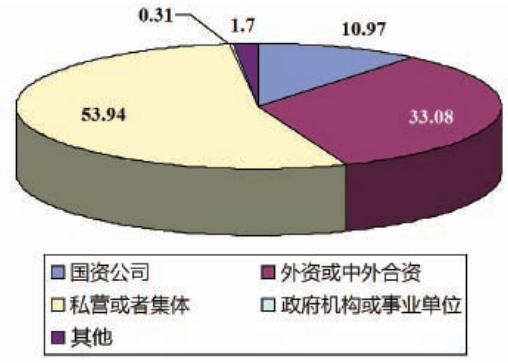


图2 国内产品管理所在企业性质分布

从业人员素质良莠不齐

就目前情况来看，国内还非常缺乏优秀的产品管理者，尤其是一线产品管理工作人员，例如产品经理、品牌经理等。造成这种情况的原因除了时间的问题外，还有以下三个原因。

■ **企业认知不足**：PMS的建立和运转涉及整个企业架构，这就要求对PMS的认识，不但高层和从业者要清晰，而且相关业务部门也必须能够清晰认知，否则PMS将无法良好运作。

■ **缺乏相应理论体系**：PMS基于“Marketing”学科范畴，但并不说明PMS的理论就是和“Marketing”的理论是一致的。就如同飞机生产一样，飞机生产本身是属于“航空制造”这门学科的，但飞机制造涉及的学科绝不仅仅限于此，还涉及到“材料学”、“空气动力学”、“流体物理学”等，只有这些学科的发展，才能从整体上促进“航空制造”的进步。

同样，PMS在80年的发展中，一方面，随着Marketing的发展，其基础知识越来越丰富；另一方面，随着市场对PMS的要求，其知识层面也越来越扩展到原来本不相关的领域，逐步开始形成一门独立的学科——产品管理学。

但就目前情况来看，这门学科离形成独立的体系并完善还有很长的路要走。虽然国外有了一些这方面的书籍，但整体看来，还是以介绍经验为主，缺乏必要的体系。

■ **缺乏足够的人才培养机制**：PMS体系再有效，缺乏高素质的职业人队伍来执行也是没有用的。就目前情况来看，大部分产品管理从业者，都是由企业原有的“市场”或者“技术”人员转型而来，再加上企业没有一套完善的培养体系，这类人才的发展“举步维艰”，甚至好多从业者到现在，都不明白产品管理到底是做什么的。一方面是企业感慨“没有高素质的产品管理专业人员”，另一方面是“大部分从业者产品管理能力不足”，该矛盾在2~3年内将持续存在。

PMS现状解决之道

既然知道了问题所在，接下来就是如何去解决问题，改变现状。可以归纳为下面四种。

认知模糊

这里的认知有两层含义：第一是企业在采用PMS之前不能客观评估自身的实际情况，找不到最合适自己的PMS形式，只是模仿类似企业；第二是对PMS本身没有清晰的认识，不知道如何引进、建设和运转，不知道需要制定什么样的规范和流程，不知道找什么样的人，如何来配置人员等，这一系列问题如果在采用PMS之前没有搞清楚，即使上了PMS，也发挥不出作用。

要解决这个问题，根本在于企业的真正态度。企业是真正希望通过采用PMS来让自身长久有利地发展，还是赶时髦，盲目跟风，看到别人采用了PMS，也跟着上，这绝对是个问题！

因此，企业在上PMS之前，一定要想清楚上PMS到底是为了什么！在上PMS之前，企业可以先问自己三个问题。

■ **我是否对现在的企业生产流程感到满意**？

■ **我是否知道现在的企业生产流程有哪些问题**？

■ **我是否能够承受上PMS的成本**？（**这些成本包括财力、阻力、传统势力**）

如果对这三个问题都持非常明确的“肯定”态度，那么，你就进入到具体的PMS准备过程中了，这就需要做好以下四点工作。

■ **全面评估企业现状**：包括目前架构、人员素质、业务流程、部门态度等。对现状进行了评估，才能做到“心中有数”，才能知道如果上PMS的话，最大的阻力可能来源于哪里。

■ **PMS咨询**：凡是要上PMS的，可以这么说，企业内部熟悉PMS的人几乎没有，因此，这时就需要企业出一些“血”，依靠“外力”，主要就是向从事PMS研究的机构进行咨询，让他们提出适合的建议作为参考。因为要

知道，PMS的体系是多样化的，企业采用什么样的体系最合适，不是凭空想出来的。

在这个阶段，既不要凭空想象，也不要完全参照咨询建议。其实，作为企业来说，既然要上PMS，就应该花些时间或者成立一个改进小组来做一些内部知识积累的工作，这是非常有必要的。否则就是“两眼一抹黑”，什么都不清楚。

- **人员准备**：需要多少人？每个人的角色是什么样的？如何进行配置？这都需要想清楚。这些决定后，就要看企业是否能够满足人员要求。刚才提到企业最好有一个改进小组，到这个阶段，改进小组就可以作为人员首选库，如果发现无法满足，就需要考虑外聘。
- **知识准备**：体系形式、人员都到位了，接下来就是准备相应的规范、流程和制度。一方面可以通过改进小组的工作来完成，另一方面可以参考类似企业的内容，并结合自身的情况进行改进，但无论如何做，只有一个原则：可执行。

以上四点并不能完全说明只要按照这个去做，就万事无忧了。建立PMS容易，但运转并且最终完善则是个长期的过程，在这个过程中，甚至会出现各种各样的摩擦。作为企业，一定要有这方面的思想准备。

从根本上说，如果企业在建立PMS上，认知非常清晰，态度非常明确，决心非常坚定，那么PMS在企业发展已经没有了一半的阻力，不存在“上还是不上PMS”的疑问，而是“如何把PMS运转好”的具体工作了。

理论欠缺

刚才提到理论欠缺最直接的影响就是“企业和个人都不知道做什么、怎么做”。

一个体系的发展离不开直接理论的支持，虽然Marketing作为PMS的理论源泉，为PMS的发展奠定了基础，但现在PMS越来越表现出更强的扩展能力，已经开始汲取其他学科的养分，而逐步开始形成一门独立的学科。

但现实的情况前面已经提到，在这个学科形成的过程中，将出现一个理论空白期，即PMS所涉及的其他学科的精华内容会大量出现，但没有进行有机的整合，直接表现是杂乱、无序和没有体系。

而这个形成过程，必定是长期的。要解决这个问题，并不是一个人、一个组织能够完成的。

理论来源于实践，然后再经过实践，经历“去伪存真”的过程，实现理论的一次升华，在一次次螺旋式的经历后，完成整合，形成独立的学科体系，并随着客观环境的变化而继续修正。

国内PMS几乎都是照搬国外，但国外企业的情况和国内有很大差异。因此，国外企业贡献的实践而形成的PMS思想不一定适合国内企业，这就说明，国内PMS理论体系的形成，根本要的是国内企业/个人的实践贡献。

在这个问题的解决上，需要三个方面的力量来共同协作才可实现。

- **企业/个人**：提供实践。
- **机构**：对实践进行理论化。
- **平台**：实现企业/个人和机构的交流渠道。

体系缺陷

这里“体系”特指“产品管理从业者的职业发展体系”，而不是企业具体的产品管理体系。

一个职业的发展，不但要有现实的从业者，而且要有足够的“后备力量”，就如同MBA有学历教育、IT从业者有职业教育一样，产品管理从业者也有有自己的“黄浦军校”才能保证PMS在国内长期发展下去。

但现实情况是：现实从业者感到“渺茫”，即将从业者感到“慌忙”，潜在从业者感到“迷茫”。

因此，要解决这个问题，需要做的工作主要包括四个方面。

- **明确定义，制订“由低到高”的发展策略**。

凡是产品管理者，都知道一句话“产品经理是没有权力的小CEO”，但请问哪位有过这样的感觉呢？

这个形容其实包含了两层含义：一方面说明了产品经理的根本工作态度；另一方面说明了产品经理的最终职业方向。

第一点暂且不说，只说第二点。这句话确实没有说错，在国外，产品经理确实是作为公司CEO的首选人群，例如微软的鲍尔默就是宝洁的产品经理出身。产品经理如果希望自己以职

业经理人的定位来发展，做到头，就一定是公司的CEO，并且这个CEO绝对不会受到行业的限制，因为在产品经理成长过程中，磨炼的是基于Marketing的产品管理思想，而这种思想相对于其他职位，是受行业限制最小的。因此，在国外，可以看到一个化妆品行业的CEO去跳槽管理一家软件企业，这是很正常的。但在国内，就不行了，很少有跨行如此之大的。

基于这点，就要求必须有一个体系化的、由低到高的产品管理人员的职业发展路线来为他们指明方向。

CPRC做过一个调查，对目前全行业的产品管理从业人员的产品管理从业时间【注：产品管理从业时间=从业时间−非产品管理从业时间】进行了统计，如图3所示。

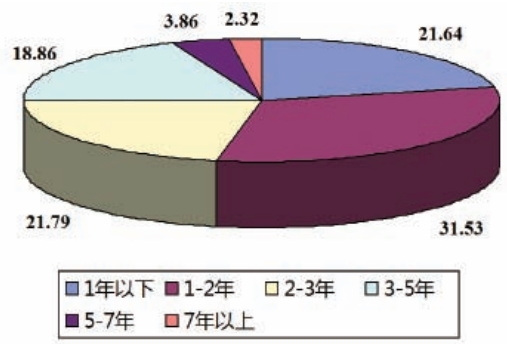


图3　国内产品管理者产品管理工作年限分布

从图3可以看出，接近80%的从业者产品管理从业时间在3年以下，而国外普遍认为培养一个优秀的产品经理，至少需要这个人在行业内摸爬滚打10年、产品管理工作时间不能少于5年。

同时可以看出，53.17%的产品管理者正处于“摸着石头过河，并且正处于河当中”的阶段，如果有一个系统的职业发展路线来协助他们，那么这些人极有可能在2~3年内成为国内第一批真正意义上的产品管理者；但如果仍然缺乏这样的路线，那么，相信在他们当中，会有很多人由于看不到明确的方向和路线而就此退出这个职业。

因此，要根据现阶段的从业者实际特征来进行职业发展路线的设计，并且根据不同阶段从业的需求提供相应的帮助。只有这样，才能让广大的产品管理人员知道“我要去哪里，我该怎么走，我现在在哪里”。

■ **建立专业的职业教育机构**。

方向有了，就需要专业的职业教育机构来协助他们走好每一步。即使短时间内无法做到类似于PMP这样标准的职业认证，但专业的职业教育还是非常有必要的。不可否认，目前在国内，也有了一些从事产品管理领域培训的机构，但就从其设计的课程来看，存在以下问题。

没有针对性：对外宣称是“产品管理”培训，但实际上依然是一个大杂烩。课程中的任何一个知识点都不是产品管理本身的内容，也就是说，这样的课程，产品管理者可以听，市场经理、项目经理同样可以听。这充分说明该机构根本没有真正明白产品管理的含义是什么。

理论多，经验少：现在国内PMS是需要“实践多于理论”的阶段，尤其是对那些3年以下的产品管理从业者，他们对理论的需求远远没有对经验的需求大，而有些课程只是一味地宣讲理论，照搬书本，大部分内容告诉你“是什么”，而很少告诉你“如何做”，对于听课者来说，就是“听的时候热血沸腾，用的时候束手无策”。

不结合国情：国内有一批以“海归”为主的产品管理培训机构，不否认他们在国外企业的产品管理经验确实很丰富，但如果只是把这些经验不加本地化地传递给国内的“土鳖”，作为“土鳖”来说，习惯了“大米面条”的肚子，一下子要装进如此多的“面包奶酪”，估计没有几个会消化良好的。

滥竽充数：这类培训机构是最可恶的，他们的出现，就是对整个体系的破坏，不但浪费了从业者的金钱和时间，更为严重的是会误导从业者的职业方向。

■ **企业要有内部培训机制和明确的责、权、利定义**。

作为企业员工，都希望能够在企业中得到素质的全面提高。而产品管理者，作为企业中的精英群体更是如此，并且产品管理者素质的提升，也是为企业的持续发展在储备后备力量和新鲜血液，是个双赢的事情。

因此，企业需要定期对产品管理人员进行职业教育，具体的内容可以根据企业的期望进行设计，可以是理论提高，也可以是经验交流，形式也可以多样化，在此就不做太多说明了。

下面重点说一下产品管理者“责、权、

利”的定义。

这是一个在产品管理人群中讨论最多的话题，摆开来说，就是许多从业者希望企业能够明确地告诉他们：我的职责是什么；我有多少权力来完成职责；完成以后，我能够获得什么。

看似一个很简单的需求，其实在许多企业中，并没有一个明确的说明。而正是这种不明确，对产品管理者的心理造成了一定程度的影响，进而影响到具体的工作。

责、权、利，这三者是统一的，而有些企业在强调责任时，却不能给予相应的权力和个人的满足，或者有些企业只是一味地许诺高薪，但却和个人工作职责关系不大。无论何种情况，对于整个PMS的发展而言，都是不健康的。

图4展示的是美国硅谷产品经理的平均薪资水平。差距之大，显而易见，而造成这种差距的直接原因就是“责、权、利”的不统一。把产品经理的工作职责人为缩小，肯定就不能给予相应的待遇，而这种人为的因素，根本来说，是对产品管理和产品管理者的模糊认识造成的。

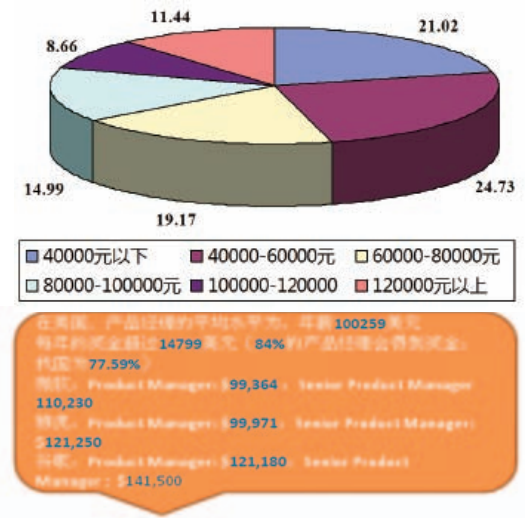


图4　美国硅谷产品经理的平均薪资水平

个人素养

即使前三个问题都解决了，而作为产品管理从业者，不去发挥主观能动性，有意识地提高自己的职业素养，那么作为条件的前三个因素将无法发挥作用。

大家普遍认为：产品管理者应该是一个“杂家”，凡是和PMS流程有关的知识，产品管理者都需要了解。

那么，这些知识都包括什么呢？按照CPRC定义的产品管理人才模型（如图5所示），可以分为三个类别的内容。

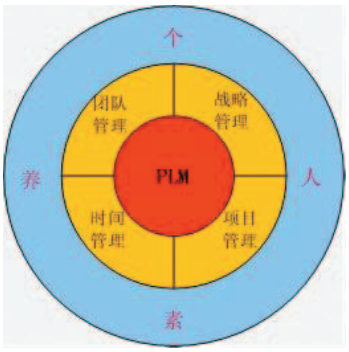


图5　CPRC定义的产品管理人才模型

个人素养：包括个人修养、创新能力、沟通协调能力、自我管理能力、工作压力承受能力。

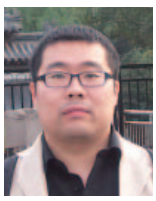
管理知识：包括战略管理知识、项目管理知识、时间管理知识、团队管理知识。

产品管理：包括产品规范管理、产品需求管理、产品项目管理、产品生命周期管理、产品品牌管理。

以上是作为一个产品管理者，需要提高的个人素养和涉及到的管理知识，具体到每一类下的工作方法，都会有所不同。因此，作为产品管理者，一方面要学习产品战略层面的知识，另一方面，还要学习产品战术执行的方法。只有掌握了这些知识，才能对自己的工作起到促进作用。

如上所说，即是对现在国内PMS的现实问题提出的解决思路。问题存在是很正常的，尤其是对于进入国内时间不长、近几年才开始兴起的PMS来说，问题肯定是会多于机会的。

但对于企业、个人、机构来说，唯一要做的工作就是实事求是地去面对这些问题，不可回避不管，也不可蛮不在乎，只要找到了问题之所在，就会找到解决之方法，随之而来的必定是光明的未来。



张智渊

UCPM中国产品经理联盟发起人，长期关注中国产品管理体系的建设和发展，曾在国内多家知名企业从事产品管理工作，具有丰富的一线产品管理经验和扎实的理论基础。

责任编辑：董世晓（dongsx@csdn.net）

如何使员工能力和收入相匹配



高阳
中国惠普资深技术顾问，微软最有价值专家（MVP）

公司与员工的价值观吻合是王道

在公司的各种要素中，最重要、最关键的是人。公司利润来自于员工的努力，公司的发展来自于员工工作能力的提高。由于每个人都需要一定的物质基础作为生活保障，需要不断提高自我能力实现更高的人生价值，如果公司能够满足员工物质层次和精神层次的需要，员工就会更加主动和积极地参与工作。

为了使员工的能力和收入相匹配，我们主要分三个层面来做。

公司层面

公司应该为员工提供良好的学习平台。一个卓越的公司，必定有一个优秀的团队和一批出色的员工。这就要求公司提供一个公平的发展平台和学习平台，让员工不断地接受新信息，学习新知识，掌握新技能，全面提高员工的综合素质和工作能力。

为员工创造良好的知识和技能交流环境。个人的视野有限，通过互相交流，每个人都可以获得全面而深刻的认识。使各种知识和能力都能得到无障碍的交流和传播，这样才更有利于学习的进步和能力的提高。

要完善工作能力和收入水平相匹配的制度。一般来说，个人的收入水平和工作能力成正比，但有些公司由于制度不够合理和完善，有的岗位不能享受公平的待遇，这将会打击员工的积极性和上进心，使无能者滥竽充数，必会给公司带来重大损失。因此，公司要完善培训制度、考核制度、薪酬制度等相关方面的制度。

管理层面

管理人员应该具有个性化的非物质引导力。积极的引导因素不仅包括工资、奖金，也包括机会、职权、信息分享、学习发展、沟通等多元要素，同时引导要素也随着个人的发展提升和不断变化。单独物质的引导力往往只能起到一时而短暂的效果。所以，管理者应该学会利用公司的职业发展平台，为员工指出职业通道，能够让他们清晰地看到未来。有更多的发展空间，能看得见、摸得着，而且他们升上去能够成功。

管理人员应该具有管理的时空艺术。比如像那种驱动型的人，他肯定是一个要挑战、有目标的人，我们就会给他挑战性的工作；而那些性格温和的人，能踏踏实实把事情做得特别漂亮，但不喜欢往前冲，需要的是和谐的人际关系，那我们就给他提供一个稳定的工作。

管理人员要承担引导员工上进的责任，并且能把公司的战略发展目标和个人的发展目标结合起来。员工能够长期保持积极向上的心态，其实更多的是职业生涯的规划。管理人员的核心能力是领导力，领导力的核心就是懂得如何激励和引导你的员工，如何关心你的员工，如何跟员工做他的职业生涯发展规划，以及他个人的行动计划，而且是

要做3~5年的计划。让更多的是员工们能看到一个机会，看到一种氛围，或者说一种体制、一种目标感。

员工层面

对于员工来说，应该清楚自己的个人职业发展规划和公司的职位发展通道，找到适合自己的方向和目标。俗话说：“人往高处走，水往低处流”，水居于下而为众人所用，人应该设定目标才能积极向上。如果想早日实现目标，就要找到自己内部驱动力（内因），自动、自发、积极主动地做事。另外，还必须要了解对于不同的职位都有哪些相应阶段的综合能力要求（能力模型），做到这些能力与收入的匹配也就是水到渠成的事情了。

从公司健康发展的角度来看，实现员工能力和收入相匹配对于这三个层面缺一不可。公司与员工的价值观吻合在一起的，由此所建立的体制就保证了所有管理者都得按这种基本理念做事，充分地信任员工，增强员工忠诚度降低跳槽风险，从而保证企业持续不断的发展。



刘舒
联想移动互联集团高级项目经理

收入与能力问题关乎企业的发展

从管理视角来看待员工能力与其获得的收入问题，有两个层面的考虑：一方面，作为企业或团队领导，是希望通过一种有效的工资及绩效考评机制，提升员工的工作满意度和稳定性，为企业的发展做贡献；而另一方面，作为企业或团队员工，是希望企业能够提供一个平台，在工作中不断提升自己的业务能力以及工资收入。可以看到，讨论员工收入与能力问题的基础，是企业发展的问题。结合我多年在各大软件及互联网企业的任职及管理经历，谈谈企业发展、员工能力以及员工收入这三者之间的关系。

企业定位要与人才模型相匹配

企业或团队定位与人才模型相匹配，是讨论员工收入与能力匹配的前提。

企业或团队首先需要明确自身的战略和业务定位，进而清晰的定义出需要的人才模型，选用适合的员工。忽略了这一点，会带来极大的风险。当初我在管理腾讯的一个测试团队时，本着能力强的原则招聘员工。而我一直笃信优秀的开发人员才能做好测试，因此选拔的对象都是能力出众的开发人员。由于腾讯能给出相对丰厚的待遇，我顺利招聘到了一名优秀的开发工程师进入测试团队。进入团队后，一方面，他发现测试团队中的开发工作没有挑战性，工作提不起兴致；另一方面，对除开发以外的测试工作，对于他而言毫无兴趣。最后过了没有半年，他就离职了。反倒是团队中几名开发能力不如他，但各方面比较均衡的复合型员工，工作完成的非常出色。从这个例子可以看到，企业或团队在开始设计考核激励机制之前，一定要思考清楚和自身业务和发展相适合的人才模型。

绩效考核和激励机制

公平、公证的绩效考核机制和富有诱惑力的激励机制，是促进员工不断提升工作能力的保障。

员工绩效，是企业或团队考核员工能力的结果。而绩效的好坏又直接和员工收入挂钩。因此，一个公平、公证的绩效考核机制，是保障员工能力与收入匹配的关键。

量化客观的绩效考核，是公平、公证的重要基础。比如，用每千行代码的缺陷率来作为一个指标，就能够相对公平、公证的考核开发工程师的代码能力。另外，核心员工的稳定性，对一个企业的发展而言，具有重要的意义。通过公平、公证的绩效考核机制识别出公司的核心员工，然后需要制订富有诱惑力的激励机制，才能稳定住这类员工。需要认识到，员工的能力和其收入不应该是线性的，而应该是指数级正相关的。一些成功的互联网企业，比如腾讯和阿里巴巴等，一旦认可员工的能力后，会不惜血本的给予其大量现金或股权的激励，使得这部分优秀人才将企业的前景和自身的前途联系在一起，作为公司基业长青的基础。



高博
盛大创新院高级研究员

加强基础学习，锻炼业务表达

让我们一开始就简化这个问题：第一，我的讨论范围仅限于程序员（包括开发工程师、测试工程师、项目经理等直接参与软件生产的职业）；第二，假定员工已经具备了一定的工作能力，着重讨论一下，在以前提下应该如何有效地使自己的收入与之相匹配。随着员工能力的提高和经验的积累，收入一般而言是肯定会多多少少随之增长的，问题的关键在于如何在同等能力条件下，避免过多的能力转换为收入方面的损失。

我的观点是：程序员要加强基础学习、锻炼业务表达。这两大法宝，既是保证现期收入的有效途径，也是打破未来收入瓶颈的必要准备。

加强基础学习

加强基础学习是最重要也是最容易被忽视的一点。程序员的工作量往往比较大，很容易习惯于Ad-Hoc方式应付手头的问题，并养成适应解决当下问题的工作模式。而在大部分程序员的工作中，使用高级数据结构和算法的机会，以及深入复杂软硬件系统底层的机会并不多。很多人在工作几年以后，在学校学习的基础知识基本上全还给了老师。

然而基础知识的掌握却是任何有价值的工作机会的核心要求，而且工作机会的价值越高，对于基础知识的掌握就是要求得全面、深入。这背后的原因也很简单：对基础知识的理解反映了程序员对其工作内容能否举一反三，是否有潜力去学习和掌握更有挑战的工作。书中自有黄金屋，这句话现在仍然成立。找回那些已经落了一层灰的《算法导论》和《编译原理》吧，你会发现结合自己的工作内容来读这些书，在精神享受的同时，也为自己的收入加上了最可靠的保险。

锻炼业务表达

锻炼业务表达是一项秘密武器，即使说它是程序员收入的一条分水岭也不为过。是的，我说的就是字面的意思：业务表达就是你能不能以最让你的客户省力的方式给出你的业务结果。让客户的加工最小化，就是给客户带来的价值最大化。

举个最简单的例子，比如要写一份缺陷报告，如果开发人员看了以后立刻知道如何复现和定位，项目经理看了以后立刻知道是否会导致设计变更，老板看了以后立刻知道是否会使产品发布延期，那么这就是一份好的报告。在做任何业务输出之前，养成反复问自己“潜在的客户有哪些”的习惯。没有人喜欢那些给自己带来额外工作量的人，多在表达上下功夫，是使得自己的能力得到充分认可的前提。当然这样做的连带效应，就是收入和能力可持续的匹配。

寻找机遇 创造未来 CSDN 热门职位全新推荐

杭州优朋电子商务有限公司

杭州优朋电子商务有限公司位于风景秀丽的美丽之洲——杭州余杭，是一家高新技术企业，专业从事计算机软件技术研发、信息技术服务及旅游电子商务网络运营。公司独立开发运营的旅游电子商务平台，为国内提供一站式服务的个性化旅游服务平台。

因业务发展需要，急需以下人才加盟，让我们一起来创造美好的明天！

- .NET高级软件开发工程师
- .NET中级软件开发工程师
- Web前端开发工程师
- 网站运营总监

简历投递：zhuqr@uniboncn.com　　联系QQ：5462240

欧特克软件（中国）有限公司

你一展身手的舞台，打造美好未来

Autodesk

加入我们的队伍，以一种全新的方式实现设计理念。无论是设计可持续性建筑还是开发混合动力汽车，Autodesk软件都可助你一臂之力。作为全球前端的二维与三维设计、工程和娱乐软件提供商，我们致力于帮助打造出更美好的未来。

让我们共同创造未来!

更多职位信息请浏览网站：www.autodesk.com/careers
或直接将简历发送至：acrd.hiring@autodesk.com

搜狐畅游

游戏研发部3D程序专员

工作职责：
1. 3D框架下各种功能模块和算法的实现；
2. 研究、开发和优化3D引擎。

职位要求：
1. 诚信正直，具备高度的责任心；2. 计算机相关专业本科及以上学历；3. 两年以上的3D引擎开发经验；4. 精通C++、面向对象编程、Windows编程；5. 熟悉3D图形学原理，熟悉3D几何；6. 精通DirectX/OpenGL，精通DX9；7. 具备良好的算法和数学基础、优秀编码习惯和学习能力。

公司网址：http://www.changyou.com
E-mail：gamejob@job.cyou-inc.com

聚胜万合信息技术（上海）有限公司

聚胜万合MediaV依托强大的互联网广告平台，为广告主、网站主和战略合作伙伴提供专业的互联网营销服务。现热招云计算工程师、PHP工程师、数据处理应用工程师、产品经理等职位，邀您共谋中国数字营销大变局。如果您想在互联网广告领域一展鸿图，如果您拥有足够的经验和才能，MediaV期待您的加入。

公司网址：http://www.mediav.com

方正国际软件有限公司

方正国际大客户业务部基于多年的自主创新积累，与Microsoft、Yahoo！、Pfizer、EMC以及ChemAxon、CLC bio等众多客户与合作伙伴在技术领域广泛而深入地合作，致力于提供技术先进、质量一流、适应能力强的BPM、ECM、MES、虚拟化、云计算和外包服务，帮助客户提升核心竞争力。

为适应业务的蓬勃发展，现在苏州、上海、北京、广州等地诚聘：

.NET/SharePoint开发人员、架构师和项目经理

简历投递：srhr1@founder.com
公司主页：www.founderinternational.com

厦门梦加网络科技有限公司

客户端程序员 （重点职位） 若干

至少一年相关工作经验，熟悉C/S架构，拥有良好的学习能力和独立解决问题的能力。根据游戏策划和主程序的要求，负责游戏客户端编程和维护工作。

招聘要求：
1. 计算机相关专业，熟练掌握Java或C#编程语言，热爱游戏编程；
2. 使用过Unity3D引擎开发过游戏者优先；
3. 公司统一使用英文版本的技术软件，拥有大量外籍员工，英语口语较好者优先。

公司地址：厦门市思明区厦禾路666号海翼大厦A座11层
公司主页：www.mengjiagames.com
公司电话：0592-2661183骆小姐　　简历投递：hr@mechanist.co

完美世界（北京）网络技术有限公司

完美世界（北京）网络技术有限公司（原完美时空）是中国领先的网络游戏开发商和运营商之一。公司成立于2004年，一直致力于创造优质的互动娱乐产业品牌，倾力打造拥有自主知识产权的高质量网游精品。

现诚聘

- ERP EBS 技术顾问
- flash（as3）开发工程师
- Java开发工程师
- 高级客户端开发工程师
- 游戏运维工程师
- PHP开发工程师
- 游戏服务器端软件工程师
- 游戏客户端软件工程师

职位详情请点击：http://special.zhaopin.com/bj/2011/wanmei03162823/index.htm
公司网址：www.wanmei.com　**简历投递：**zhaopin@wanmei.com

天津港保税区科技发展局

东软（天津）、展讯（天津）高薪诚聘IT英才

招聘职位：高级Java工程师，高级C++工程师，DSP工程师，芯片驱动工程师，应用软件工程师，驱动软件工程师，算法工程师

工作地点：

天津空港经济区（www.tjftz.gov.cn）

简历投递：resume@adm.tjftz.gov.cn

北京法国电信研发中心

Software engineer in the web service system development

Position requirements:

- Master degree in telecommunication, Compute Science OR equivalent
- Experience in J2EE, SOA, Spring, Hibernate, Struts, XML, REST, HTML5 development
- Experience in cloud computing development, such as SaaS, PaaS, etc.
- Strong programming skills, at least 3+ years experience in Java software development
- Familiar with the software development methodology
- Familiar with networking technology
- Good English communication skill
- Good team working spirit and communication skills, and hard working as well

Contact：hr@orange-ftgroup.com.cn

北京神舟航天软件技术有限公司

神舟软件是由中国航天科技集团公司控股的重点高新技术软件企业，公司业务领域涵盖工业软件与服务、电子政务与公共服务、基础软件、系统集成、航天文化创意等方面，拥有一批具有自主知识产权的实用型软件产品，产品与服务遍及机械制造、医疗卫生、教育等多个领域。

因发展需要诚征人才，我们会提供广阔的发展空间和完善的保障。

- 高级PLM业务架构师
- 高级Java系统架构师
- 高级Java开发工程师
- 项目经理/产品经理
- UI架构师
- 中级Java开发工程师

更多职位信息可浏览：http://www.bjsasc.com/rlzyavidm.html
简历投递：zhaomengyi@bjsasc.com　**联系人：赵梦祎**

爱立信（中国）通信有限公司

爱立信专注于通信研发130余年，始终屹立无线通信领域行业前沿的地位。而其多媒体软件系统更是独树一帜，成为爱立信的主要战略方向。爱立信公司多媒体事业部上海的研发中心正肩负这样的使命，其中的TV（包括IPTV、WebTV、CableTV和MobileTV）产品线更是战略重点，为此我们诚邀您加盟。

我们这次主要招聘三类职位：

1. MobileTV/WebTV的软件开发职位
2. TV解决方案的集成和测试
3. TV机顶盒的集成和测试

广州瀚信通信科技股份有限公司

招聘职位：

- C/C++系统架构师
- C/C++网络开发工程师
- Silverlight/WPF 前端交互设计师
- C#开发工程师
- 软件测试工程师

详细信息请登录CSDN网站查询！

请以“职位+姓名+CSDN”作为邮件标题，
简历投递：yfhr@hantele.com　**公司主页：**www.hantele.com

MapReduce Hold不住？

浅谈流式计算系统

文 / 杨栋

本文系统地介绍和分析比较了业界主流的Yahoo！S4、StreamBase和Borealis三种流式计算系统，希望读者能从这些系统的设计中领悟到不同场景下流式计算所要解决的关键问题。

背景

非实时计算几乎都基于MapReduce计算框架，但MapReduce并不是万能的。对于搜索应用环境中的某些现实问题，MapReduce并不能很好地解决问题。

商用搜索引擎，像Google、Bing和Yahoo!等，通常在用户查询响应中提供结构化的Web结果，同时也插入基于流量的点击付费模式的文本广告。为了在页面上最佳位置展现最相关的广告，通过一些算法来动态估算给定上下文中一个广告被点击的可能性。上下文可能包括用户偏好、地理位置、历史查询、历史点击等信息。一个主搜索引擎可能每秒钟处理成千上万次查询，每个页面都可能会包含多个广告。为了及时处理用户反馈，需要一个低延迟、可扩展、高可靠的处理引擎。然而，对于这些实时性要求很高的应用，尽管MapReduce作了实时性改进，但仍很难稳定地满足应用需求。因为Hadoop为批处理作了高度优化，MapReduce系统典型地通过调度批量任务来操作静态数据；而流式计算的典型范式之一是不确定数据速率的事件流流入系统，系统处理能力必须与事件流量匹配，或者通过近似算法等方法优雅降级，通常称为负载分流（load-shedding）。当然，除了负载分流，流式计算的容错处理等机制也和批处理计算不尽相同。

最近Facebook在Sigmod 11上发表了利用HBase/Hadoop进行实时数据处理的论文，通过一些实时性改造，让批处理计算平台也具备实时计算的能力。这类基于MapReduce进行流式处理的方案有三个主要缺点。

- 将输入数据分隔成固定大小的片段，再由MapReduce平台处理，缺点在于处理延迟与数据片段的长度、初始化处理任务的开销成正比。小的分段会降低延迟，增加附加开销，并且分段之间的依赖管理更加复杂（例如一个分段可能会需要前一个分段的信息）；反之，大的分段会增加延迟。最优的分段大小取决于具体应用。
- 为了支持流式处理，MapReduce需要被改造成Pipeline的模式，而不是Reduce直接输出；考虑到效率，中间结果最好只保存在内存中等。这些改动使得原有的MapReduce框架的复杂度大大增加，不利于系统的维护和扩展。
- 用户被迫使用MapReduce的接口来定义流式作业，这使得用户程序的可伸缩性降低。

综上所述，流式处理的模式决定了要和批处理使用非常不同的架构，试图搭建一个既适合流式计算又适合批处理计算的通用平台，结果可能会是一个高度复杂的系统，并且最终系统可能对两种计算都不理想。

目前流式计算是业界研究的一个热点，最近Twitter、LinkedIn等公司相继开源了流式计算系统Storm、Kafka等，加上Yahoo!之前开源的S4，流式计算研究在互联网领域持续升温。不过流式计算并非最近几年才开始研究，传统行业像金融领域等很早就已经在使用流式计算系统，比较知名的有StreamBase、Borealis等。

本文简单介绍几种业界使用的流式计算系统，希望流式系统的设计者或开发者们能从中获得启示。

图1从整个分析系统的架构角度，给出了实时计算子系统所处的位置。实时计算系统和批处

理计算系统同属于计算这个大的范畴，批处理计算可以是MapReduce、MPI、SCOPE等，实时计算可以是S4、Storm等，批处理和实时都可以或不依赖统一的资源调度系统。另外，计算系统的输入、输出，包括中间过程的输入、输出，都与存储系统交互，可以是块存储系统HDFS，也可以是K-V存储系统Hypertable等。计算层的上层是数据仓库，或者直接和用户交互，交互方式可以是SQL-like或者MR-like等。

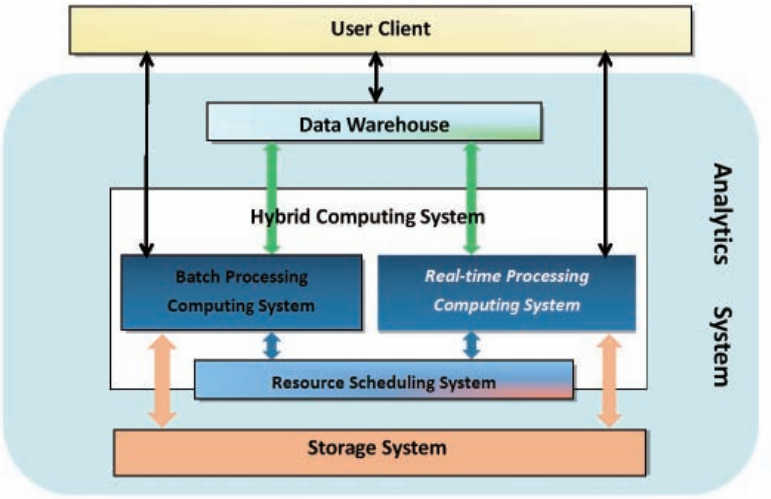


图1 数据分析系统整体组成示意图

系统

S4

S4是一个通用的、分布式的、可扩展的、分区容错的、可插拔的流式系统。基于S4框架，开发者可以轻松开发面向持续流数据处理的应用。

S4的设计特点有以下几个方面。

■ Actor Model

为了能在普通机型构成的集群上进行分布式处理，并且集群内部不使用共享内存，S4架构采用了Actor模式，这种模式提供了封装和地址透明语义，因此在允许应用大规模并发的同时，也提供了简单的编程接口。S4系统通过处理单元（Processing Elements，PEs）进行计算，消息在处理单元间以数据事件的形式传送，PE消费事件，发出一个或多个可能被其他PE处理的事件，或者直接发布结果。每个PE的状态对于其他PE不可见，PE之间唯一的交互模式就是发出事件和消费事件。框架提供了路由事件到合适的PE和创建新PE实例的功能。S4的设计模式符合封装和地址透明的特性。

■ Decentralized and Symmetric Architecture

除了遵循Actor模式，S4也参照了MapReduce模式。为了简化部署和运维，从而达到更好地稳定性和扩展性，S4采用了对等架构，集群中的所有处理节点都是等同的，没有中心控制。这种架构将使得集群的扩展性很好，处理节点的总数理论上无上限；同时，S4将没有单点容错的问题。

■ Pluggable Architecture

S4系统使用Java开发，采用了极富层次的模块化编程，每个通用功能点都尽量抽象出来作为通用模块，而且尽可能让各模块实现可定制化。

■ Partial Fault-Tolerance

基于Zookeeper服务的集群管理层将会自动路由事件从失效节点到其他节点。除非显式保存到持久性存储，否则节点故障时，节点上处理事件的状态会丢失。

■ Object Oriented

节点间通信采用“Plain Old Java Objects”（POJOs）模式，应用开发者不需要写Schemas或用哈希表来在节点间发送Tuples。

S4的功能组件分3大类，Clients、Adapters和PNode Cluster，图2显示了S4系统框架。

S4提供Client Adapter，允许第三方客户端向S4集群发送事件和接收事件。Adapter实现了基于JSON的API，支持多语言实现的客户端驱动。

Client通过Driver组件与Adapter进行交互，Adapter也是一个Cluster，其中有多个Adapter结点，Client可以通过多个Driver与多个Adapter进行通信，这样可以保证单个Client在分发大数据量时Adapter不会成为瓶颈，也可以确保系统支持多个Client应用并发执行的快速、高效和可靠性。

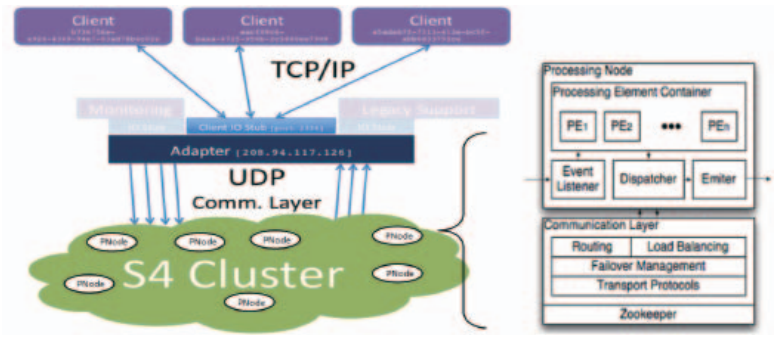


图2 Yahoo！S4流式系统框架结构图

在Adapter中，真正与Client交互的是其Stub组件，该组件实现了管理Client与Adapter之间通过TCP/IP协议进行通信的功能。其中，GenericJsonClientStub这个类支持将事件在Client与Adapter之间以JSON的形式转换，从而支持更多种类型的Client应用。不同的Client可以配置不同的Stub来与Adapter进行通信，用户可以定义自己的Stub来实现自己想要的业务逻辑，这样也使得Client的行为更加多样性、个性化。

StreamBase

StreamBase是IBM开发的一款商业流式计算系统，在金融行业和政府部门使用，其本身是商业应用软件，但提供了Develop Edition。相对于付费使用的Enterprise Edition，前者的功能更少，但这并不妨碍我们从外部使用和API接口来对StreamBase本身进行分析。

StreamBase使用Java开发，IDE是基于Eclipse进行二次开发，功能非常强大。StreamBase也提供了相当多的Operator、Functor以及其他组件来帮助构建应用程序。用户只需要通过IDE拖拉控件，然后关联一下，设置好传输的Schema并且设置一下控件计算过程，就可以编译出一个高效处理的流式应用程序了。同时，StreamBase还提供了类SQL语言来描述计算过程。

StreamBase的组件交互情况如图3所示。

StreamBase Server是节点上启动的管理进程，它负责管理节点上Container的实例，每个Container通过Adapter获得输入，交给应用逻辑进行计算，然后通过Adapter进行输出。各个Container相互连接，形成一个计算流图。

Adapter负责与异构输入或输出交互，源或目的地可能包括CSV文件、JDBC、JMS、Simulation（StreamBase提供的流产生模拟器）或用户定制。

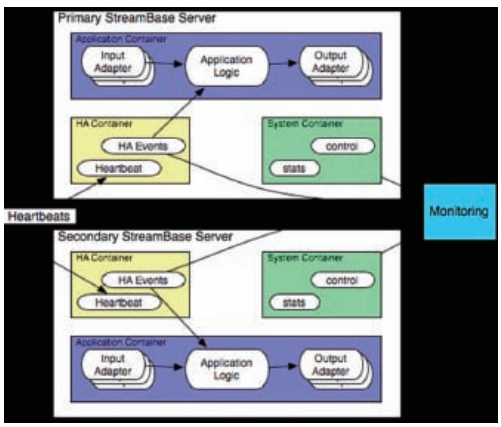


图3 StreamBase组件交互图

每个StreamBase Server上面都会存在一个Sytsem Container，主要是产生系统监控信息的流式数据。

HA Container用于容错恢复，可以看出它实际包含两个部分：Heartbeat和HA Events，其中HeartBeat也是Tuple在Container之间传输。在HA方案下，HA Container监控Primary Server的活动情况，然后将这些信息转换成为HA Events交给StreamBase Monitor来处理。

Monitor就是从System Container和HA Container中获取数据并且进行处理。StreamBase认为HA 问题应该通过CEP方式处理，也就是说如果哪个部件出现问题，就肯定会反映在System Container和HA Container的输出流上面，然后Monitor通过复杂事件处理这些Tuples的话就能够检测到机器故障等问题，并作出相应处理。

StreamBase提出了以下4种模板策略来解决容错问题。

Hot-Hot Server Pair Template

Primary Server和Secondary Server都在同时计算，并且将计算结果交给下游。优点是Primary Server如果故障的话那么Secondary Server依然工作，几乎没有任何切换时间；并且下游只需要选取先到来的Tuple就可以处理了，保证处理速度最快；缺点是浪费计算和网络资源。

Hot-Warm Server Pair Template

Primary Server和Secondary Server都在同时计算，但只有Primary Server将计算结果交给下游。优点是如果Primary Server故障，Secondary Server可以很快切换，而不需要任何恢复状态的工作。相对于Hot-Hot方式时间稍微长一些，但没有Hot-Hot那么耗费网络资源，同时也浪费了计算资源。

Shared Disk Template

Primary Server在计算之后，将计算的一些中间关键状态存储到磁盘、SAN（Storage Area Network）或是可靠的存储介质。如果Srimary Server故障，Secondary Server会从介质中读取出关键状态，然后接着继续计算。优点是没有浪费任何计算和网路资源，但恢复时间依赖状态的量

级而定，相对于前两种，恢复时间可能会稍长。

■ Fast Restart Template

这种方案限定了应用场景，只针对无状态的应用。对于无状态的情况，方案可以非常简单，只要发现Primary Server故障，Secondary Server立即启动，并接着上游的数据流继续计算即可。

Borealis

Borealis是Brandeis University、Brown University和MIT合作开发的一个分布式流式系统，由之前的流式系统Aurora、Medusa演化而来。目前Borealis系统已经停止维护，最新的Release版本停止在2008年。

Borealis具有丰富的论文、完整的用户/开发者文档，系统是C++实现的，运行于x86-based Linux平台。系统是开源的，同时使用了较多的第三方开源组件，包括用于查询语言翻译的ANTLR、C++的网络编程框架库NMSTL等。

Borealis系统的流式模型和其他流式系统基本一致：接受多元的数据流和输出，为了容错，采用确定性计算，对于容错性要求高的系统，会对输入流使用算子进行定序。

Borealis的系统架构如图4所示。

■ Query Processor（QP）是计算执行的地方，是系统的核心部件，其大部分功能继承自Aurora。

■ I/O Queues将数据流导入QP，路由Tuples到其他节点或客户端程序。

■ Admin模块用来控制本地的QP，例如建立查询、迁移数据流图片段，该模块也会同Local Optimizer协作优化现有数据流图。

■ Local Optimizer职责包括本地调度策略、调整Operator行为、超载后丢弃低价值元组等。

■ Storage Manager模块用于存储本地计算的状态数据。

■ Local Catalog存储本地数据流图和元数据，可以被本地所有组件访问。

■ Borealis Node还有彼此通信的模块用于执行协作任务。

■ Neighborhood Optimizer使用本地和邻居节点来优化节点间的负载均衡或shed load。

■ High Availability （HA）模块相互监测，发现对方故障时及时代替对方。

■ Local Monitor收集本地性能相关统计数字

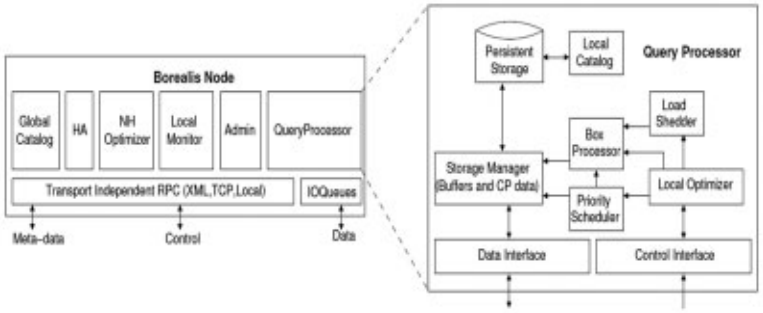


图4 Borealis框架示意图

报告给本地和Neighborhood Optimizer。

■ Global Catalog为整个数据流计算提供了一个逻辑上的完整视图。

除作为基本功能节点外，Borealis Server也可以被设计成一个协作节点来执行全局的系统监控和其他优化任务，比如全局的负载分布和Global Load Shedding，因此Borealis实际上提供了完整的3级监控和优化（Local、Neighborhood、Global）。

负载均衡方面，Borealis提供了动态和静态两种部署机制。

■ Correlation-based Operator Distribution

通过分析不同Operators和Nodes间的负载变化的关系，决定和动态调整Operatpr的部署，使之达到负载均衡。

■ Resilient Operator Distribution Algorithm

该算法的目标是提供一种静态的Operator部署方案，该方案能够在不需要重新调整的情况下处理最大可能的输入速度变化范围。

由于动态调整需要时间和消耗，前者适用于负载变化持续时间较长的系统；而后者则能处理较快较短的负载峰值。在实现上前者使用相关系数作为节点关联度指标，并通过贪婪算法将NP问题转化为多项式求解；而后者在部署前计算完毕，保证系统能够容忍负载峰值。该算法在线性代数上建模，包括Operator Ordering、Operator Assignment两个阶段。

Borealis通过四种容错机制来满足用户需求。

■ Amnesia Backup

备机发现主机故障，立即从一个空的状态开始重做。

■ Passive Standby

主机处理，备机待命，主机按周期做Checkpoint，主机故障后切换到备机，重放Checkpoint和数据流，对于不确定性计算可以很

表1　3种流式系统的模型对比

Models			S4	StreamBase	Borealis
Processing Model	Select Policy		Program.	Multiple	Program
	Consume Policy		Program.	Zero	Zero
	Load Shedding		No	Yes	Yes
Deployment Model			Clustered	Clustered	Clustered
Interaction Model	Observation		Push	Push	Push
	Forwarding		Push	Push /Pull	Push
	Notification		Push	Push	Push
Time Model			Stream-only	Stream-only	Stream-only
Rules Model	Transforming		Explicitly	Implicitly	Implicitly
Data Model	Nature of Items		Event	Data	Data
	Format		Objects	Tuples	Tuples
	Support for Uncertainty		No	No	No
	Nature of Flows		Homo	Homo	Homo
Function Model	Clock		Present	Present	Absent
	Sequence		Bounded	Unbounded	Unbounded
	Knowledge Base		Present	Present	Present
	Dynamic Rules Change		No	No	No
Language Model	Single-item	Selection	No		
		Projection	No		
		Renaming	No		No
	Logic	Conjunction	No		No
		Disjunction	No		No
		Repetition	No	No	No
		Negation	No		No
	Sequence		No	No	No
	Iteration		No	No	No
	Windows	Fixed			No
		Landmark			No
		Sliding		No	
		Pane	No		No
		Tumble	No		
		User Defined	No		No
	Flow Management	Join			
		Union	No		
		Except	No		No
		Intersect	No		No
		Remove Dup	No	No	No
		Duplicate	No	No	
		Group By	No		
		Order By	No		No
	Parameterization		No		
	Flow Creation		No	No	No
	Aggregate				

好地支持，缺点是恢复时间较长。

■ Active Standby

主备机同时计算，主机故障时直接切换到备机，不支持不确定性计算，浪费计算资源，不过恢复时间几乎没有。

■ Upstream Backup

通过上游备份来容错，故障时从上游重放数据即可，恢复时间最长，不过最节省资源。

除此之外，Borealis还提供了更高级的容错机制Rollback Recovery，它是一种基于副本在节点失效、网络失效或网络分区时的故障恢复机制，在尽量减少系统不一致的情况下，尽可能地保证系统的可用性。该机制允许用户定义一个阈值来在一致性和可用性之间做一个平衡。当系统数据恢复后，系统支持重新计算输出正确的结果，保证最终一致性。该机制使用了Data-serializing Operator（SUnion）来确保所有的副本处理同样顺序的数据。当失效恢复后，通过Checkpoint/Redo、Undo/Redo来实现恢复重放。

对比

表1就上述3个流式系统做个分类比较，比较项基于DEBS2011会议上IFPSurvey中涉及的各种Models。Processing Model描述流元组进行计算时的选择策略、消费策略及负载降级处理。Interaction Model描述输入组件和计算系统、计算系统内部及计算系统和输出组件的交互方式。Time Model描述事件流是否按照时间约束。Rules Model描述流式计算规则是显示还是隐式。Data Model描述流中的数据组成、格式等。Function Model描述流式计算系统的功能模型。Language Model描述语言层面的各种算子。

小结

本文介绍了业界主流的3个流式计算系统，希望从这些系统的设计中领悟到不同场景下流式计算所要解决的关键问题。

Yahoo! S4的最新版本是Alpha version v0.3.0，动态负载均衡和在线服务迁移等重要功能都尚未实现，不过其代表性的3个特点值得学习，Actor模式、非中心化的对称结构及可插入式的架构。

StreamBase是有着功能强大的IDE并且支持控件式的方法来搭建应用程序，同时还提供了高级语言来搭建应用程序的方法。由于是商业产品，其用户接口的精彩设计值得借鉴，同时其可组合的HA方案也是亮点之一。

Borealis是学术界研究的重要产出，它对新一代的流式系统涉及的诸多方面，如系数据模型、负载管理、高可用性、可扩展性都作了全面和翔实的研究，一方面系统变得强大、先进，另一方面使得系统也变得臃肿、复杂。这套系统的许多策略都值得我们学习，可以应用于不同的流式计算场景。



杨栋

百度分布式高级研发工程师，从事Hypertable、Hadoop及流式计算的研究和开发。

责任编辑：董世晓（dongsx@csdn.net）

海量数据分析的“双立人”：YunTable

文 / 吴朱华

身为一家初创公司，很难一下子像大公司那样为客户海量数据分析相关的需求做“一桌子菜”，而在分布式数据库基础上引入MapReduce编程模型做成的YunTable，是一个专注于百台级别的服务器集群和PB级别数据量的分布式数据库，可以称得上是一把海量数据分析的“双立人”。

现在无论是企业还是个人，它们产生的数据量不断增大，而且递增速度非常惊人。根据统计数据显示，超过50%的企业目前每日生成的数据量在1TB以上，超过10TB的有10%，甚至有5%的企业每日生成的数据量已经达到了50TB以上。

海量数据的管理已成为企业非常头疼的问题，并且很多公司都希望能对这些海量数据进行深入和细致地分析，来获取第一手的商业资料，这样能对企业的运行做出正确的判断，同时这样的需求不仅来自新兴的互联网企业，很多大中型的传统企业也有类似的需求。

2011年5月，麦肯锡全球研究院（MGI）发布了一篇名为“*Big Data: The next frontier for innovation, competition, and productivity*”的报告【注：报告链接为http://www.mckinsey.com/mgi/publications/big_data/pdfs/MGI_big_data_full_report.pdf】，在这份报告中，充分肯定了海量数据的价值。

虽然海量数据的市场前景很大，但我们身为一家初创公司，很难一下子像大公司那样为客户海量数据分析相关的需求做“一桌子菜”，也就是提供一整套完善的、基于客户业务的、全面定制的海量数据分析平台，所以我们团队专注于提供用于海量数据分析的“厨具”。做好这件“厨具”绝对是一门学问，其中如何选好方向更是如此。

MapReduce与分布式数据库之争

在谈到海量数据的工具方面，现在主要是MapReduce和分布式数据库这两种模式之间的争斗。对于初创团队而言，选择一个合适和有长远发展潜力的技术切入点极为关键。当然在这点上，我们曾经也像很多用户那样，在这两种技术方向之间徘徊。

MapReduce的优点在于灵活性，因为能够非常方便地使用成熟的程序语言（比如Java或者Python等）来编写较复杂的逻辑，而且特别善于处理类似网页这样的半结构化数据，以及类似PDF电子文档这样的非结构化数据。MapReduce的最著名产品就是开源项目Hadoop。

分布式数据库的优点在于已经有几十年技术的积累，比如在数据压缩和列存储等方面，其工作负载的性能与MapReduce相比有较明显的优势；在接口方面，分布式数据库能支持SQL这样比较完善和有一定积累的规范；更重要的是在数据的可管理性方面，分布式数据库也有其天生的优势，知名的产品有EMC的Greenplum和Teradata等。

我们团队的关注点在于结构化数据。由于

结构化在管理和处理方面有明显的优势，对数据进行结构化非常有意义，所以我们认为如果有些数据用户没有对其进行结构化的想法，那就说明这类数据本身并不是非常有价值。基于此，我们认为分布式数据库在结构化方面的优势是我们团队坚持和发扬的方向，并且将在此基础上引入MapReduce这样灵活的编程模型来争取做一把用于海量数据分析的“双立人”。下面就给大家介绍一下这把“双立人”，也就是YunTable。

产品介绍

从2010年起，我们就开始了YunTable的研发工作，在2011年6月完成了YunTable 1.1 版的开发，同时与多家合作伙伴一起推进YunTable，并将于下半年投入正式的商业运行。

YunTable是专注于百台级别的服务器集群和PB级别左右数据量的分布式数据库，在架构上引入了很多经典的Google BigTable设计，并提供一整套完善的海量数据分析功能，包括支持大多数常用SQL指令和MapReduce编程模型，让中型互联网企业和大中型传统企业用户能在较低成本的情况下获得一个用于海量数据分析的私有云。从整体而言，YunTable功能方面比较接近EMC的Greenplum，但更着重于伸缩性和整体成本。

核心目标

在产品特色方面，YunTable主要专注于以下四个方面。

- **伸缩性强：** 由于YunTable的很多设计都继承自BigTable，比如“轻Master”等，所以在伸缩性方面非常不错，使得用户能够非常简单地通过向一个YunTable集群添加新的服务器来线性提升整个集群的数据存储和分析能力，并且整个集群的规模能延伸至PB级别和百台以上的服务器。
- **强大的海量数据分析能力：** 在海量数据的情况下，能够以并行的方式将数据极快地导入其中，并且为大多数数据处理相关的SQL命令提供出色和可计算的速度，同时性能将随着负载实例的增多而不断提升，而且提供了MapReduce这样灵活的编程模型。
- **数据安全性：** 引入了数据备份和自动恢复等机制，确保用户的数据不会出现任何丢失的情况。
- **整体成本低：** 无需昂贵的小型机或者高端的x86服务器，只需多台普通的x86服务器就能搭建一个数据存储集群，并简化整个集群的管理，帮助用户降低用于海量数据处理的总体拥有成本（TCO）。

整体架构

如图1所示， YunTable的整体架构主要由Master节点、Region节点和Client端三种模块组成。

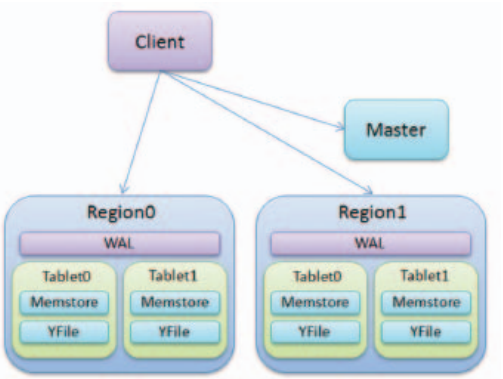


图1 YunTable的整体架构

Master节点

Master节点在功能上面属于比较“轻”的，主要用于管理整个集群，并负责维护Table和Region节点之间的对应关系，实际数据的查询和输入则大部分通过Region节点和Client端之间的交互完成，与Master节点无关，以此降低整个Master节点的负载，从而能有效管理百台级别的集群。

Region节点

Region的作用是负责处理来自Client端的请求，并存储和管理大量的数据。一般而言，推荐一个节点对应一个CPU核、一块网卡和一块SATA硬盘，所以一台普通的x86服务器能运行多个Region节点。在结构方面，首先，Region

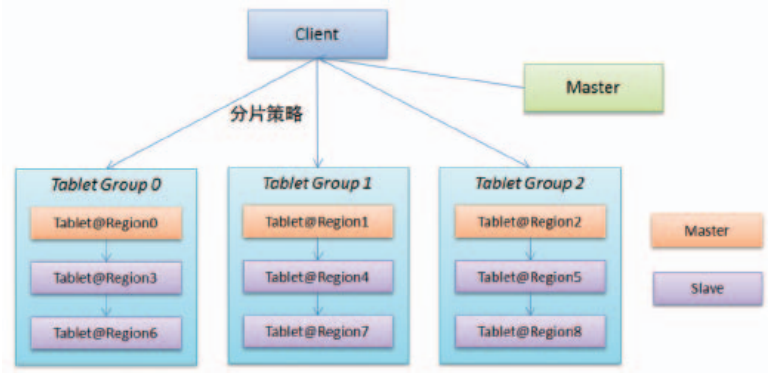


图2 YunTable的分布式设计

节点有WAL（Write-Ahead Logging，预写式日志），通过这个日志可以顺序记录写入的数据，来保证数据的安全性，接着数据会按照其所属的Table存储到对应的Tablet中，而Tablet则作为一个容器来存储对应的Table在这个Region上的所有数据。在一个Tablet中，最新的热点数据会放置在一个运行在内存中的Memstore，接着达到一定时间或者Memstore的存储量到达一定的阈值时，Memstore中的数据会批量写入到一个YFile中。Region节点采用了数据压缩、SSTable和行列混合存储等技术。

Client端

Client端主要有两类，一类是名为YunCli的主要用于管理集群和用于快速导入海量数据的命令行界面；另一类是基于JDBC规范的Java驱动，主要用于常规SQL命令的执行，这对于大部分企业应用而言，是非常友好的。

分布式设计

在分布式架构方面，当然基本Region、Master和Client这三者关系是不变的，但由于我们所面对的是海量数据的存储环境，所以无法对Table大小进行限定，也就是一个Table的大小有可能远超过一台机器所能承载的容量，同时也希望将一个Table数据尽可能地分发到多个Region节点上面来将海量数据处理尽可能地并行化，再加上一个Table需要有多个分布在不同机器上的备份来保证数据的安全性，所以如图2所示，我们引入了Tablet Group（Tablet组）机制。

一个Table包含多个Tablet Group，每个Tablet Group将根据用户选择具体的分片策略来存储一个Table中的部分数据，具体Tablet Group的数目由用户自己设置，并且每个Tablet Group之间中存储的数据是不重复的。为了数据的安全性，每个Tablet Group会包括多个Tablet，具体数目主要由用户设定的备份（Duplication）值来确定，并将其中第一个Tablet设为Master，而将其余的Tablet设为Slave。

在Tablet Group之间数据的备份参考了经典的Master/Slave机制，也就是由Master Tablet来接收新数据的写入，并写入到硬盘上的WAL日志，接着Master Tablet将这些新数据异步发给其第一个Slave Tablet，然后Slave Tablet之间会进行数据的接力。

对于那些对实时性很高的简单查询，它们可以通过只访问所有Tablet Group中的Master Tablet来保证数据读写的强一致性；但对于那些数据一致性和实时性要求较低的数据分析命令，可以通过并发访问所有Tablet来提升性能。

在分片策略方面，YunTable主要支持以下三种。

- **负载均衡**：顾名思义，在这种策略下，系统会自动地把数据尽可能均匀地分配到每个Tablet Group中。这种策略的优点是简单，可以让用户在基本没有什么设置的情况下，轻松地将数据均衡地分配到每个节点上，这对于那些需要经常涉及到类似全表查询的场景而言是非常有益处的，因为它能最有效率地利用每台服务器的I/O性能，但对部分数据的实时查询而言，其性能将不会是最佳情况。
- **手动分片**：用户在创建数据表的时候，选取主键或者某一个栏位作为Distributed Column（分布栏位），接着针对于这个Distributed Column所负责的范围手动设置每个Tablet Group。这种方式的好处是在数据均匀分配的情况下，不仅能保证类似全表查询，而且对于部分数据的实时查询也非常有帮助，但在创建数据表时，需要花一定时间去规划。当然在实际运行时，用户也能继续对具体分片进行调整，并且在手动分片部分将支持基于Hash的算法。

■ **智能分片**：这是在手动分片基础上的提升，用户在设置手动分布时可以激活这个特性，通过该特性，能够将上面手动调整的工作来进行自动化，但如何用好智能分片，用户需要学习。

虽然这三种分片机制各有不同，但其实质都是为了今后的应用尽可能将数据均匀分布，用户在具体使用时，应基于实际应用场景来确定选择哪种分片策略。

YunTable 1.1

YunTable 1.1在2010年6月上旬完成相关的开发工作，在功能方面，YunTable 1.1支持大多数基于单表的SQL命令，包括基本的CRUD（增加、查询、更新和删除）、用于数据分析的Group By与DISTINCT等命令，以及COUNT、MIN、MAX和SUM等函数。

6月中旬，根据合作伙伴提供的用于国家电网的测试用例，我们进行了关于YunTable 1.1的测试，这个测试主要涉及到一张存储海量个人用户电力使用数据的表，其共有11列数据。我们对此表数据容量达1亿条和5亿条这两个情况进行了测试，主要测试的是数据装载和全表查询这两个功能。

首先，在数据装载方面，在本次测试中，一个小型的YunTable集群（3台服务器，5个实例）每秒稳定在4.3万行数据左右。但在测试中发现，这4.3万行只是测试程序能生成的，并不是YunTable的瓶颈，所以估计一个小型YunTable集群每秒中能导入10万~20万行数据，这应该能满足大多数海量数据场景的需求，同时经过算法的压缩，整个数据所占用的面积仅为未压缩之前的十分之一。

其次，对于全表查询，在2007年主流服务器硬件的基础上，一个YunTable实例能够在30秒左右完成对1亿行的全表查询，并且YunTable扩展性极佳，一个有5个实例的小型YunTable集群能在30秒左右的时间完成对5亿行的全表查询，也就是说，假设有25台普通x86服务器，每台机器4个YunTable实例，那么就能在1分钟内完成对1张有100亿行表的全表查询，而现阶段很少有超过100亿行的表，这个性能以及伸缩性应该在业界属于比较领先的。

YunTable 2.0

在YunTable 2.0版中，我们将继续重点强化其数据分析能力，主要是以下两个方面。

其一是提供对多表Join（链接）的支持。虽然现阶段有些用户使用反范式化（Denormalization）的方式来完成对海量数据的多维分析，但如果我们的产品能够对这个特性有很好的支持的话，将会方便用户做类似的操作，尽管多表Join有可能不像全表查询那样是YunTable的核心竞争力之一。

其二是提供MapReduce的支持。这有点类似传统的UDF（User Defined Function，用户自定义函数），通过这个功能，能让用户编写复杂的脚本来灵活地满足业务的需求，将MapReduce的灵活性和分布式数据库强大的性能结合在一起，让结构化数据处理更方便和更快捷。

最后，希望能在与用户的合作中进一步提炼和完善YunTable，争取成为用于海量数据分析的“双立人”，利用其“锋利”的功能来有效地对PB级别以上的海量数据进行快速和接近实时的分析。



吴朱华

《云计算核心技术剖析》一书作者，曾在IBM中国研究院参与多款云计算产品的研发，现为人云科技创始人兼总经理，海量数据分析数据库YunTable和PaaS系统YunEngine核心架构师。

责任编辑：董世晓（dongsx@csdn.net）

云计算新兴领域——安全即服务

文 / Tim Mather, Subra Kumaraswamy, Shahed Latif　　译 / 刘戈舟，杨泽明，刘宝旭

安全即服务（Security as a Service）作为云计算的一个新兴领域，已有一些云计算服务提供商采用自己定制的云计算环境为多个机构提供安全即服务了，但还没有正规的、有实力的云计算服务提供商将安全即服务作为一种集成产品提供出来。

与软件即服务（Software as a Service）相似，安全即服务（Security as a Service）的业务模式也是基于订购的。

对于安全即服务，有两种新兴的提供商模式：第一种包括已有的信息安全提供商，它们通过包含在云计算中的交付服务对原有的交付模式进行了更改；第二种包括新成立的信息安全公司，这些公司是新兴的纯粹云计算服务提供商——也就是说，这些公司提供的安全只作为云计算服务，并不提供传统的客户端/服务器端模式的网络安全、主机安全以及/或者应用程序安全产品。

在那些更改业务模式以涵盖安全即服务的已有信息安全公司中，最引人注目的是传统的反恶意软件提供商。然而，其他已有的信息安全公司也提供安全即服务的交付，特别是对于电子邮件的过滤。

起源

安全即服务的起源有三个推动力。最早的推动力在距今十年前：垃圾邮件，或者说是不明来源的电子邮件。早在1999年，就有公司【注：例如Postini公司，它于2007年9月被Google公司收购】提供如下的电子邮件服务。

Postini公司的理念是：电子邮件应当变得更好，而且可以变得更好。尽管电子邮件是最流行的互联网资源，但服务提供商和软件开发者并没有使电子邮件变得更好，最糟糕的是，积极的市场营销人士把任何电子邮件用户都作为潜在的用户。Postini公司的服务可以用来扩展你的服务提供商的电子邮件产品的能力。针对垃圾邮件的服务只是第一步。在未来的几个月中，你会看到更多的Postini服务，使电子邮件变得更好【注：根据2000年5月10日Postini.com的互联网档案馆时光机。另一家电子邮件过滤公司MessageLabs，成立于1999年，并在2008年5月被Symantec收购，曾经把自己称为应用程序服务提供商——云计算的早期演变】。

一些其他的公司现在也提供电子邮件过滤服务，这都是些独立的安全公司，许多互联网服务提供商（ISP）也常常使用自己的品牌转售这些独立安全公司的服务。

安全即服务的第二个推动力是托管安全服务（MSS）。托管安全服务提供商（MSSP）已经为用户提供了数年的外包服务，在这种模式下，MSSP替机构管理其网络安全设备，例如，防火墙和入侵检测系统（IDS）。使用MSS的动力与云计算相同的：相对于机构内部的解决方案，通过共享资源可以降低费用。MSSP和CSP的区别在于：MSSP的共享资源是人员，而不

是基础设施。此外，由于很多机构没有配备人员去处理全天24小时对于此类服务的支持，也没有为此类岗位配置充足的具有专门知识的人员，MSSP的共享服务（如人员）模式在财政上则变得十分有吸引力。MSSP模式成为云计算服务提供商的推动力，是因为MSSP模式打破了机构信息安全项目外包部分非正式但十分强大的障碍。在这种情况下，外包也意味着信息安全设备的远程运行模式管理。（虽然外包信息安全常常是个选项，但最初的外包是趋于本地运行模式，即在用户自己的设施内，与远程运行模式有所不同。当然，现在的外包可以是本地运行模式、远程运行模式、在岸外包、离岸外包以及交付模式的其他变体。）

在MSSP模式下，虽然网络安全工作外包了，但用户安全的责任还是留在了用户这边。用户负责管理和监测MSSP，并选择执行各种安全策略。MSSP监测和管理设备（如防火墙、入侵检测系统）以及数据流（如网络、内容或电子邮件过滤），这些设备（包括用来管理和监测数据流的设备）是属于用户的。因此，节约成本和提高效率也只能到此而已。尽管这是订购类型的服务（运营成本，常写作OpEx），但还存在用户本地硬件的相关资本开支（常写作CapEx）。使用云计算，由于绝大部分的设备以及监测和管理都由安全即服务提供商负责，资本开支将进一步缩减。

安全即服务的第三个推动力是直接在终端提供安全，组织效率会下降。不仅仅因为终端数目的激增，许多配置变量也是IT部门无法有效管理的；此外，由于许多终端是可移动的，解决配置问题并保持安全软件及时更新也会成为艰巨的任务；再加上许多移动设备缺乏足够的资源（如处理能力、内存和存储能力）用以处理当今的终端保护套件，终端保护的情况并不乐观。

由于这些问题，加之恶意软件的爆炸式增长，终端保护成为一个日益严重的问题。例如，“在2008年，Symantec检测出1656227个恶意代码威胁……这占据了Symantec长期以来检测的约260万恶意代码威胁的60%”【注：Symantec全球互联网安全威胁报告：“2008年趋势”，第14卷第10页，2009年4月出版】。这导致了终端保护的思维改变。为什么不通过云计算保护终端，而舍弃在终端上进行终端保护呢？也就是说，为什么不经过云计算清理出入终端的网络流量，舍弃试图监测和管理终端自身的复杂处理，将监测和管理出入终端网络流量的活动（而不是监测和管理终端设备自身）转移到云计算中进行呢？

> 安全即服务有两种新兴的提供商模式：第一种为已有的信息安全提供商，通过包含在云计算中的交付服务对原有的交付模式进行了更改；第二种为新成立的信息安全公司，它们是新兴的纯粹云计算服务提供商。

将反恶意软件保护措施迁移到云计算中而不是驻留在终端，这个概念在2008年7月于加利福尼亚州圣何塞市召开的USENIX 会议的论文中得到了相当大的强化。论文标题为“CloudAV: N-Version Antivirus in the Network Cloud”，可于http://www.eecs.umich.edu/fjgroup/pubs/cloudav- usenix08.pdf处获取，论文表明基于云计算的反病毒（如反恶意软件）对近期威胁的检测能力较基于终端的单引擎检测能力可提高35%，使总体检测率达到98%，这显著优于运行在终端的单引擎的检测结果。（由于终端资源的限制以及运行多引擎的不兼容性，终端一般一次仅仅运行一个反恶意软件引擎。）

当今的产品

当今在安全即服务部分的产品涉及几个改进信息安全的服务：电子邮件过滤（包括备份、归档和电子发现【注：电子发现是指民事诉讼方面对于电子格式信息的发现。虽然电子发现不（也不应当）局限于电子邮件，但由于机构的许多信息是通过电子邮件传送的，电子邮件明显成为开始电子发现程序的最佳位置】，网页内容过滤，漏洞管理，身份即服务（在本文中写作IDaaS）。

电子邮件过滤

在电子邮件方面，安全即服务主要涉及清理垃圾邮件、钓鱼电子邮件以及包含在传入到机构的电子邮件流中的恶意软件，然后将干净的电子邮件安全的提供给机构，以便邮件不会再次遭受污染。这种方法的好处是：由于使用多引擎，用户不仅能得到更加全面的安全，还可以改善设备的性能（因为反恶意软件的引擎是运行在云计算中，而不是直接运行在终端），并能够得到更好的对于反恶意软件的管理。反恶意软件管理优于终端解决方案，是因为反恶意软件是与操作系统和处理器无关的，这样可以通过云计算集中管理，而不用处理很可能从多个反恶意软件提供商处获得的多个管理系统。这种在云计算中清理的服务必然会带来的好处是：减少电子邮件使用的带宽、降低机构电子邮件服务器的负荷，以及提高机构（或收件人）自身反恶意软件努力的效果。

虽然在电子邮件方面，安全即服务的绝大部分注意力集中于入站电子邮件，但在出站电子邮件方面也常常使用安全即服务。许多机构想要确保他们不会无意间发送感染了恶意代码的电子邮件，通过安全即服务清理出站电子邮件是阻止此类问题和避免尴尬的好方法。此外，对于出站的电子邮件可以使用安全即服务以执行机构关于电子邮件加密的策略（例如，在指定的电子邮件域内，如属于业务合作伙伴或者属于用户的电子邮件域）。通常电子邮件加密在（电子邮件）服务器到服务器层面执行，这样不需要个人用户参与，也不需要密钥管理。这可以通过在传输层的网络通信中使用安全套接字层（SSL）或传输层安全协议（TLS）实现。

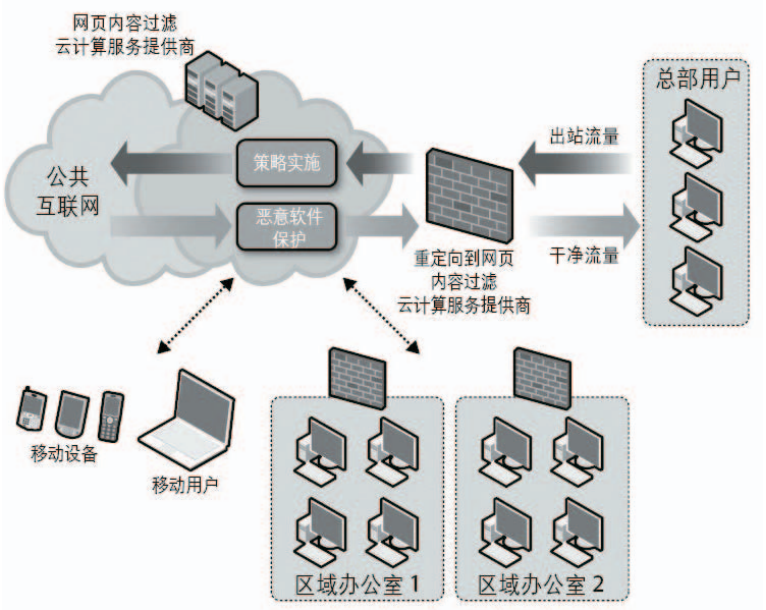


图1　安全即服务网页内容过滤

使用安全即服务反恶意软件的进一步好处是，在跨越了企业所有终端的可见恶意软件威胁当中提炼出来的整体智能特性，与类型（如服务器、台式机、笔记本电脑或移动设备）、位置、操作系统或者处理器架构无关，可以及时拥有更大的视野，对于机构的信息安全队伍而言有很大帮助。

电子邮件安全即服务也包括电子邮件备份和归档。这个服务通常涉及在集中的存储库中存储和索引机构电子邮件信息及附件。这个集中的存储库允许机构通过一些参数索引和搜索，参数包括数据范围、收件人、发件人、主题和内容。这些功能对于电子发现非常有用，如果没有这些功能，电子发现的过程可能耗资巨大。

网页内容过滤

当属于机构的终端，无论在机构设施内、在家或者在公路上，试图获取网页流量时，这些流量被转移到安全即服务提供商处，并进行恶意软件威胁的扫描，以确保只有干净的流量传送到终端用户。机构可以通过允许、阻止或限制流量（减少流量所用的带宽）的方式执行其网页流量策略。由于当今可访问网站的数目异常庞大，早期部署于机构内部的URL（统一资源定位符）过滤解决方案变得越来越低效。安全即服务提供商可以通过检查HTTP（超文本传输协议）头信息、页面内容以及嵌入的链接，更好地了解网站内容，从而为URL过滤措施提供辅助。此外，这些服务使用集体信誉记分制以加强过滤的精度。

网页内容的安全即服务还包括对出站网络流量扫描，防止用户可能没有合适授权（数据泄露保护）而向外传递敏感信息（如ID号码、信用卡信息、知识产权）。网络流量的扫描还

包括内容分析、文件类型以及模式匹配，以阻止数据泄露。

图1描述了安全即服务网页内容过滤。

漏洞管理

随着互联网机构在规模和复杂度上的增长，以及这些机构运行重要性的提升，确保相关系统的安全配置及运行变得愈加重要，也愈加困难。有的安全即服务提供商对系统漏洞进行发现，评估并确定优先级别，之后报告并补救这些漏洞以确保系统的安全运行。这些信息也用于监测及报告合规性（如支付卡行业的数据安全标准）。

身份管理即服务

相对于电子邮件过滤、网页内容过滤和漏洞管理等安全即服务中的实力型产品，身份管理即服务（IDaaS）是近期出现的安全即服务的实例。与云计算中使用的身份及访问管理（IAM）相比，目前的身份及访问管理在功能上有明显的缺陷（如可扩展性）。身份管理即服务试图在云计算中提供身份及访问管理服务。比较早期的身份管理即服务产品往往把重点放在身份验证上，因为这是用户最关心的问题，参见图2。然而，对于云计算服务提供商而言，最重大的问题是关于身份管理即服务提供商，以及开发某种形式的协同元系统。（正如元目录在机构内不会改变规模，虚拟目录也不会根据云计算的水平而改变规模。）身份管理即服务提供商也需要为云计算用户提供其他的身份及访问管理服务，包括授权（至少包括群组和角色）、用户开通和审计。

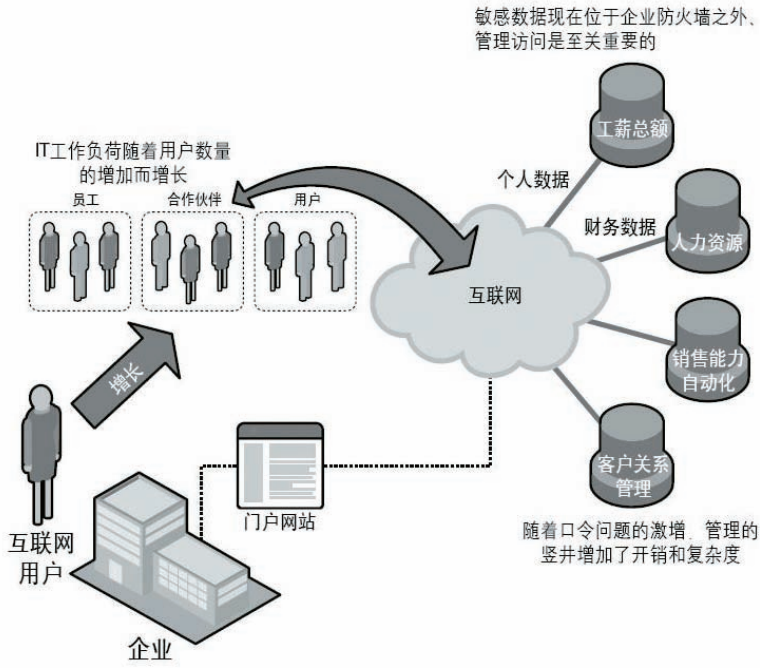


图2 身份管理即服务模式

小结

当前的几个安全即服务产品不仅是可行的，而且是相当成熟的。电子邮件和网页内容过滤已经提供了几年的时间，而且提供此类服务的方法也十分成熟和先进。一些安全即服务产品还比较新，仍处于完善过程中（如身份管理即服务）。目前，已有一些云计算服务提供商采用自己定制的云计算环境为多个机构提供安全即服务了，但还没有正规的、有实力的云计算服务提供商（包括Amazon、Google、Microsoft、Proofpoint、RightScale、Salesforce、Sun、Workday等）将安全即服务作为一种集成产品提供出来。



本文节选自机械工业出版社华章公司《云计算安全与隐私》一书。该书详述了当把数据交付给云计算时所面临的风险，以及为了保障虚拟基础设施和网络应用程序的安全可以采取的行动。特此感谢机械工业出版社授权。

架构师的武器之如何界定项目边界

文 / 王小虎

架构方法论的重要性已经毋庸置疑，传统方法已经越来越不能适应日益变化的IT架构。然而，架构方法论实践过程中遇到的挑战和陷阱依然无处不在。为此，IBM软件集团大中华区合作伙伴技术支持总经理王小虎撰写了“架构师挑战与武器”系列文章，并将根据CSDN网站上“架构师挑战”的调查投票结果撰写专稿，陆续刊登于CSDN网站。

挑战——如何界定项目边界，把握系统全局

现实挑战：新上一个IT系统，往往不是独立存在的，一般都需要与外围实体和系统进行交互，而需要集成交互的内容很多，哪些集成是本系统需要实现的呢？同时，因为时间紧张，有一些集成工作需要在第一阶段去完成，但另外还有一些工作需要放到第二阶段，怎样界定系统的边界呢？

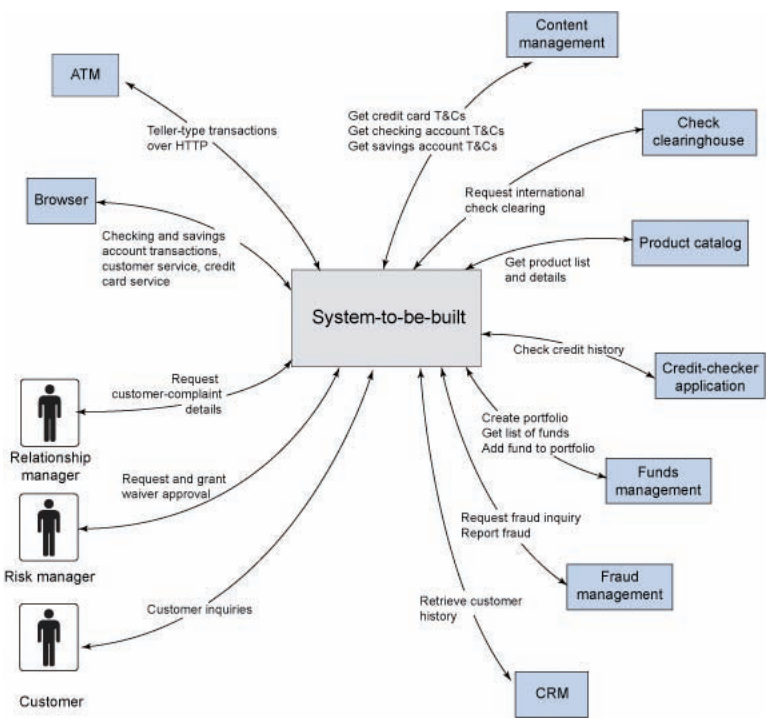


图1　银行应用程序的系统上下文关系（注：本图来源于IBM developerWorks网站）

传统的应对办法：与客户进行多次会议研讨，反复讨论，用文字零星地记录。

危害：场景一，突然有一天客户说，我们系统的客户数据应该从SAP中取得，而订单数据需要送到数据仓库中去。突然冒出来这两个新的集成任务，增加了工作量不说，另外由于没有提前协调这两个系统的实施商，还得从预算中抠出一部分费用让他们做一定的开发修改，造成项目延期、预算超支。

场景二，临上线时发现，该系统和办公系统的集成需要跨越内外网，内外网不允许直接连接，需要双方都开发新的连接组件透过网闸来连接。

传统办法的症状分析：缺乏直观正式的工作件来记录系统边界，记录系统范围的文字很多，但却很容易忽略和遗漏重要的系统以及约束。

架构师的十八般武器之一：用好系统环境图，高瞻远瞩把握全局

系统环境图是什么？系统环境图用图形结合文字的方式直观地描述系统的边界。它把系统当成一个黑盒子。让待建系统与其他需要与待建系统接受和发送信息的外部实体在一个图中展示出来，构成一幅系统级别的视图。

系统环境图长什么样子？图1显示了一个银行应用程序的系统上下文关系图例子。

- 将待构建系统表示为黑盒。
- 描述其与外部实体（系统和最终用户）的

交互。

- 标识系统和外部实体间的信息和控制流。

系统环境图的四大元素

Actor和Users

表明与系统交互的Actor和用户。

通道

用户将使用不同的通道来访问系统。具体包括通道及通常使用此通道与系统交互的角色和用户的类型、通道使用的设备、通道支持的网络和带宽，用于在系统之间发送和接收数据的访问协议。

外部系统

必须记录系统在执行所需的功能时与之交互的外部系统。

信息流

信息流是在该系统和外部系统、用户角色和通道间流动的内容。信息可以传统批量或实时方式传送。将信息及其特征作为系统环境的一部分加以记录在定义总体软件架构时极为重要。

对于当期要实现的集成，使用实线表示；对于当期不实现但是未来需要实现的集成，可以用虚线表示。

系统环境图也有多种视图

从功能层面和运行层面来看。系统环境图可以是功能层面的，描述功能层面需要集成的信息。系统环境图也可以运行层面的，描述外部系统或者组织坐落的位置、连接的方式、透过的协议。

系统环境图也可以是逻辑的和物理的。比如对于外部用户和角色，在逻辑的角度，会考虑用Actor来抽象化描述用户；在物理的角度，会用User来记录有多少用户以及他们所处的位置。

如果剔除外围系统，逻辑功能层面的系统环境图有点像最高级别的用例集合。

使用系统环境图的好处是显而易见的，包括以下内容。

- 可以快速地框定系统边界。透过系统环境图，可以表明将哪些外部实体纳入整体解决方案的范畴内，以清晰地框定系统的边界。
- 可以与客户直观地展示和讨论。如果拿着一个冗长的记录不全的文档去和客户交流，客户很容易遗漏一些外围实体、通道以及重要的信息流。
- 容易厘定责任。图形是精确化的建模方法。通过图形来圈定本次要实现的内容，避免模棱两可的文字。如果将来还需要集成新的系统，还可以和客户协商新增需求部分的费用。
- 避免未来需求和现在需求纠缠不清，对于系统能建立一个整体的概念和认识。
- 帮助梳理大的非功能需求。在系统环境图的文字部分，记录信息交互的特性。比如：将信息分类为批处理、实时或半实时类别，每个单位时间必须支持的事务信息，组成典型事务的数据类型，每个事务通常涉及的数据量，事务执行的频率。
- 指导后续设计。可以指导后续用例、组件模型和运行模型的设计。用例是系统环境图的逻辑功能层面的细化。组件模型是功能层面的细化，运行模型是物理层面的细化。
- 赢得客户的信任。使用正规的工作产品，可以更好地与客户沟通需求，可以帮助客户梳理对系统整体的认知，正规的做法让客户心里更有底，从而赢得客户的青睐和信任。

关于系统环境图的更详细的资料，可以参见developerWorks网站文章《编写软件架构文档说明，第2部分：开发系统上下文》。

备注：架构师挑战与武器系列文章之后续内容敬请关注，包括：进行合理的架构决策，重用架构资产，设计高性能系统、如何合理估算系统尺寸和资源，设计安全的架构，设计高可靠性的架构，设计灵活高效的信息架构，画架构总揽图，设计可集成的系统，找到正确的组件，正确的摆放组件，灵活的领域模型，设计可运维的系统，如何找到合理的服务，如何把握架构设计的粒度，编写合理的架构文档，评估架构可行性，规避架构设计中的风险。详情请参见CSDN网站。



王小虎

IBM软件集团大中华区合作伙伴技术支持总经理，中国IT行业架构方法论启蒙者和领导者，积极推动国内IT行业架构方法论的学习和实践，在培养国内IT企业的架构师方面作出了卓越的贡献。

Redis源代码分析(下)

文 / 阮若夷

Redis是一个开源Key-Value内存数据库，以支持丰富的数据结构著称，支持主从复制、持久化等高可用特性，可以和程序无缝结合，本文继续上期分析Redis的源代码方法，帮助大家了解这一款数据库的工作机理。

服务端处理过程

图1涵盖了接收request、处理请求、调用函数、发送reply的过程。

我们先从Redis的网络事件库开始。Redis并没有使用libevent等第三方网络框架，而是自己实现一个reactor的事件分离器，封装了epoll、select、kqueue，代码精练，只有1KB多的代码。

开启监听后，会给监听fd注册一个文件事件，当fd有数据可读状况下，执行acceptTcpHandler函数。

客户端发起连接时，从epoll_wait返回，由于监听的fd处于可读状况，调用先前注册过的acceptTcpHandler函数，产生一个连接fd，创建一个redisClient对象，并为接下来的读取数据注册一个readQueryFromClient的只读事件，把这些都存储在aeEventLoop里的events里，fd就是数组的index，clientData就是redisClient，rfileProc保存函数。处理完监听fd后，就处理这个连接fd，执行刚才注册的readQueryFromClient函数。

readQueryFromClient先从连接fd中读取数据，先存储在c→querybuf里（networking.c 823）。接下来函数processInputBuffer来解析querybuf，上面说过如果是telnet发送的裸协议数据是没有*打头的表示参数个数的辅助信息，针对telnet的数据跳到processInlineBuffer函数，而其他则通过函数processMultibulkBuffer。这两个函数的作用一样，解析c→querybuf的字符串，分解成多参数到c→argc和c→argv里面，argc表示参数的个数，argv是个redis_object的指针数组。每个指针指向一个redis_object，object的ptr里存储具体的内容，对“get a”的请求转化后，argc就是2，argv就是：

```
(gdb) p (char*)(*c->argv[0])->ptr
$28 = 0x80ea5ec "get"
(gdb) p (char*)(*c->argv[1])->ptr
$26 = 0x80e9fc4 "a"
```

协议解析后就执行命令。processCommand首先调用lookupCommand找到get对应的函数。上面讲过从redisServer→commands这个hash表查找对应函数。这里就查找到的是getGenericCommand（t_string.c 62行）。

getGenericCommand的工作原理非常简单，根据redisServer→db这个hash table查询“a”这个key，并通过applyReply函数把需要返回的数据存储在c→buf上，字符串的长度为c→bufpos。然后需要把数据写入到连接fd中，所以为这个fd添加一个可写的函数sendReplyToClient。

等到事件分离器开始处理可写事件时，就会调用sendReplyToClient函数，同样redisClient对象从文件事件的clientData里获得。sendReplyToClient函数从c→buf上把数据写到连接的fd里，发送的长度记录在c→sendlen，直到

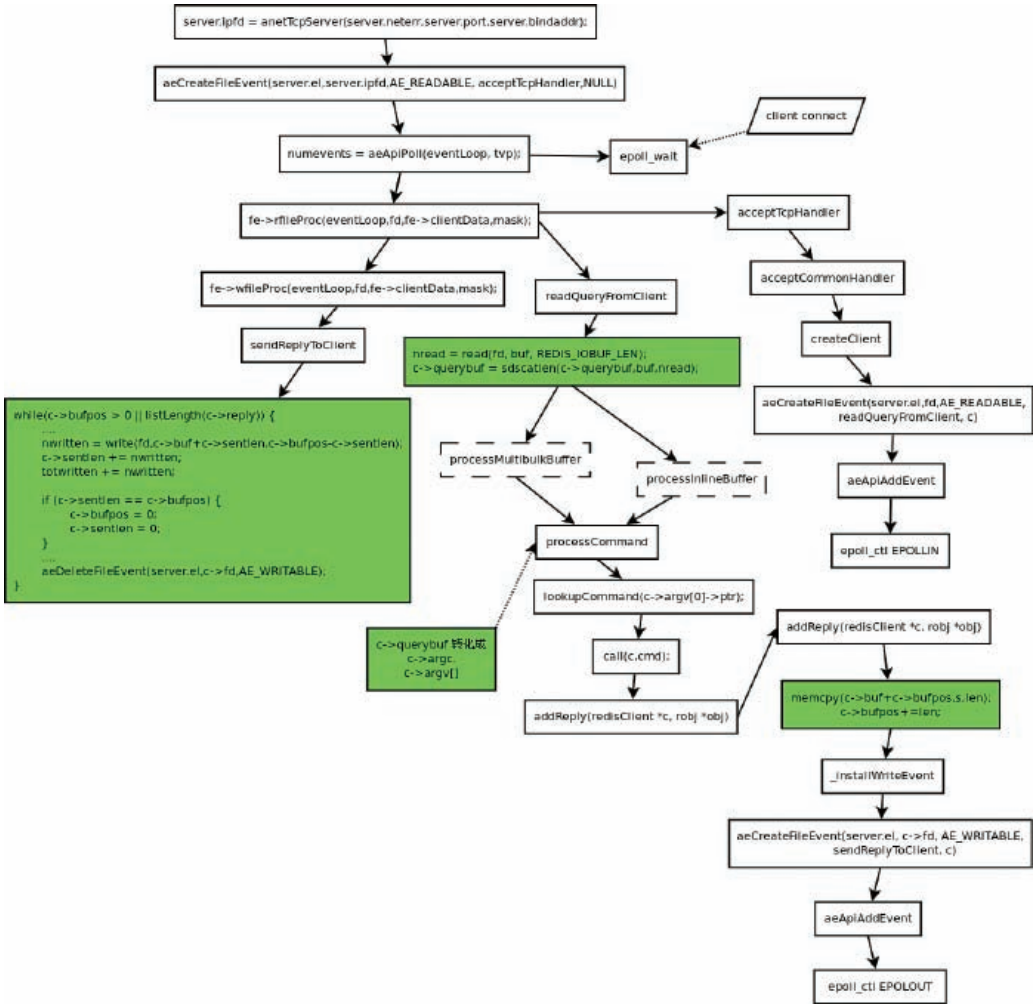


图1 接收request、处理请求、调用函数、发送reply的过程

c→bufpos==c→sendlen就可以认为buf内的数据都已经发送出去。

回复协议

下面介绍一下Redis服务端的reply协议。reply协议分很多种，有bulk replies、err message、integer reply、status reply、Multi-bulk replies。

■ bulk replies

是以$打头消息体，格式$值长度\r\n值\r\n，一般的get命令返回的结果就是这种个格式。

```
redis>get aaa
$3\r\n
bbb\r\n
```

对应的的处理函数addReplyBulk：

addReplyBulkLen(c,obj);

addReply(c,obj);

addReply(c,shared.crlf);

■ error message

以-ERR打头的消息体，后面跟着出错的信息，以\r\n结尾，针对命令出错。

```
redis>d
-ERR unknown command 'd'\r\n
```

处理的函数是addReplyError：

addReplyString(c,"-ERR ",5);

addReplyString(c,s,len);

addReplyString(c,"\r\n",2);

■ integer reply

以:打头，后面跟着数字和\r\n：

```
redis>incr a
:2\r\n
```

处理函数是：

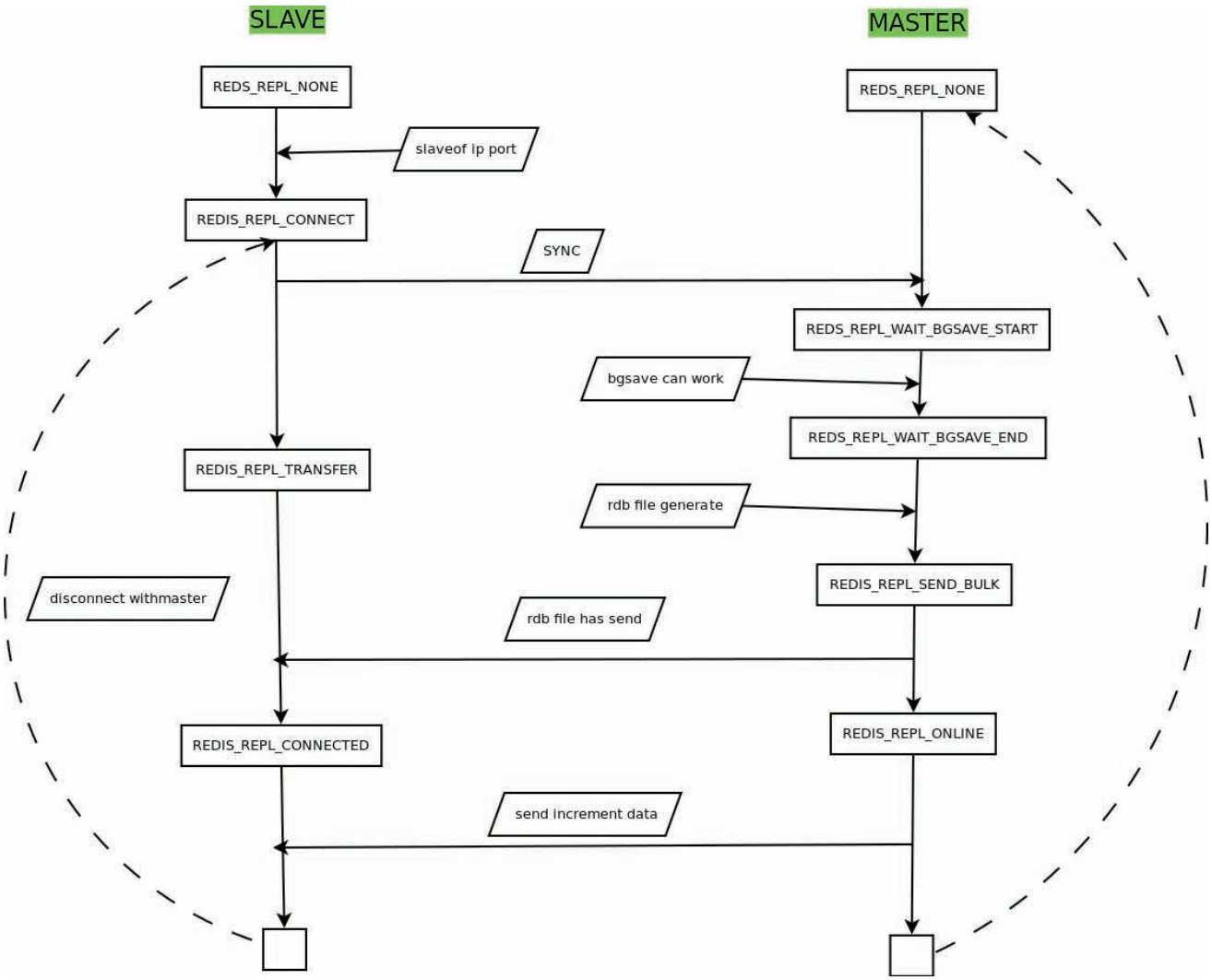


图2 Redis的复制原理

```
addReply(c,shared.colon);
addReply(c,o);
addReply(c,shared.crlf);
```

■ status reply

以+打头，后面直接跟状态内容和\r\n：

```
redis>ping
+PONG\r\n
```

这里要注意reply的内容经过协议加工后，都会先保存在c→buf里，c→bufpos表示buf的长度。待到事件分离器转到写出操作（sendReplyToClient）的时候，就把c→buf的内容写入到fd里，c→sentlen表示写出长度。当c→sentlen=c→bufpos才算写完。

■ Multi-bulk replies

lrange、hgetall这类函数通常需要返回多个值，这种消息结构与请求的格式一模一样。相关的函数是setDeferredMultiBulkLength。临时数据存储在链表c→reply里，处理方式同其他的协议格式。

复制

Redis复制的原理和使用都非常简单（如图2）。只需要在Slave端键入：

```
slaveof masterip port
```

就开启了从Master到Slave的复制。一个Master可以有多个Slave，Master有变化时会主动把命令传播给每个Slave。Slave同时可以作为其他Slave的Master，前提条件是这个Slave已经处

于稳定状态（REDIS_REPL_CONNECTED）。Slave在复制的开始阶段处于阻塞状态（sync_readline），无法对外提供服务。

取消复制，从Slave状态的转换回Master状态，切断与原Master的数据同步。

```
slaveof no one
```

Slave端接收到客户端的slaveof masterip port命令之后，调度slaveofCommand保存的masterip、port，修改server.replstate为REDIS_REPL_CONNECT，然后返回给客户端，这个复制的行为是异步的。

Slave端主线程在时间事件serverConn（redis.c 518行）里执行replicationCron（redis.c 646行）开始与Master的连接。syncWithMaster函数与Master的通信。经过校验之后（如果需要），会发送一个SYNC command给Master端，然后打开一个临时文件用于接收接下来Master发过来的rdb文件数据。再添加一个文件事件注册readSyncBulkPayload函数，这个就是接下来用于接收rdb文件的数据的函数，然后修改状态为REDIS_REPL_TRANSFER。

Master接收到SYNC command后，跳转到syncCommand函数（replication.c 556行）。syncCommand会调度rdbSaveBackground函数，启动一个子进程做一个全库的快照，并把状态改为REDIS_REPL_WAIT_BGSAVE_END。Master的主线程的serverCron会检查这个持久化的子进程是否退出。

```
if ((pid = wait3(&statloc,WNOHANG,NULL))
!= 0) {
       if (pid == server.bgsavechildpid)
{
                backgroundSaveDoneHandler
(statloc);
}
```

如果bgsave子进程正常退出，会调用backgroundSaveDoneHandler函数继续复制的工作，该函数打开刚刚产生的rdb文件。然后注册一个sendBulkToSlave函数用于发送rdb文件，状态切换至REDIS_REPL_SEND_BULK。sendBulkToSlave作用就是根据上面打开的rdb文件，读取并发送到slave端，当文件全部发送完毕之后修改状态为REDIS_REPL_ONLINE。

我们回到Slave，上面讲到Slave通过readSyncBulkPayload接收rdb数据，接收完整个rdb文件后，会清空整个数据库emptyDb()（replication.c 374）。然后就通过rdbLoad函数装载接收到的rdb文件，这样Slave和Master数据就一致了，然后把状态修改为REDIS_REPL_CONNECTED。

接下来就是Master和Slave之间增量传递的增量数据，另外Slave和Master在应用层有心跳检测（replication.c 543）和超时退出（replication.c 511）。

持久化

Redis有全量（save/bgsave）和增量（aof）的持久化命令。

全量持久化

遍历所有的RedisDB，读取每个bucket里链表的Key和Value并写入dump.rdb文件（rdb.c 405）。save命令直接调度rdbSave函数，这会阻塞主线程的工作，通常我们使用bgsave。

bgsave命令调度rdbSaveBackground函数启动了一个子进程然后调度了rdbSave函数，子进程的退出状态由serverCron的backgroundSaveDoneHandler来判断，这个在前面复制章节已经提及。

除了直接的save、bgsave命令之外，还有几个地方还调用到rdbSaveBackground和rdbSave函数。

shutdown：Redis关闭调度的prepareForShutdown会做一次持久化工作，保证重启后数据依然存在，会调用rdbSave()。

flushallCommand：清空Redis数据后，如果不做立即执行一个rdbSave()，生成一个空的快照出现crash后，可能会载入含有老数据的快照。

```
void flushallCommand(redisClient *c) {
touchWatchedKeysOnFlush(-1);
server.dirty += emptyDb();      //
清空数据
addReply(c,shared.ok);
if (server.bgsavechildpid != -1) {
       kill(server.
bgsavechildpid,SIGKILL);
       rdbRemoveTempFile(server.
bgsavechildpid);
}
rdbSave(server.dbfilename);    //
没有数据的dump.db
```

```
        server.dirty++;
}
```

sync：当Master接收到Slave发来的该命令时，会执行rdbSaveBackground，这个以前也有提过。

数据发生变化：在多少秒内出现了多少次变化则触发一次bgsave，这个可以在conf里配置。

```
for (j = 0; j < server.saveparamslen; j++)
{
        struct saveparam *sp = server.
saveparams+j;
        if (server.dirty >= sp->changes &&
now-server.lastsave > sp->seconds) {
                rdbSaveBackground(server.
dbfilename);
                break;
  }
}
```

增量持久化

aof原理有点类似redo log。每次执行命令后如出现数据变化，会调用feedAppendOnlyFile，把数据写到server.aofbuf里。

```
void call(redisClient *c, struct
redisCommand *cmd) {
        long long dirty;
        dirty = server.dirty;
        cmd->proc(c);        //执行命令
        dirty = server.dirty-dirty;
        if (server.appendonly && dirty)
                feedAppendOnlyFile(cmd,c-
>db->id,c->argv,c->argc);
```

待到下次循环的before_sleep函数会通过flushAppendOnlyFile函数把server.aofbuf里的数据write到append file里。在redis.conf里配置每次write到append file后从page cache刷新到disk的规律。

```
#appendfsync always
appendfsync everysec
#appendfsync no
```

该参数的原理MySQL的innodb_flush_log_at_trx_commit一样，是个比较影响I/O的一个参数，需要在高性能和不丢数据之间做平衡。软件的优化就是平衡的过程，没有银弹。

为什么会先写到server.aofbuf，而不是直接写到aof文件内，这是一个优化的过程，合并多次的小I/O成一次大的连续I/O。

先写到server.aofbuf，然后再写到数据文件，过程中如果crash会不会丢数据呢？

答案是不会，我们先看看网络事件库如何处理读写事件：

```
if (fe->mask & mask & AE_READABLE) {
        rfired = 1;
        fe->rfileProc(eventLoop,fd,fe-
>clientData,mask);
}
if (fe->mask & mask & AE_WRITABLE) {
if(!rfired||fe->wfileProc!=fe-
>rfileProc)
fe->wfileProc(eventLoop,fd,fe-
>clientData,mask);
}
```

rfired变量决定了在一次文件事件循环内，如果对于某个fd触发了可读事件的函数，不会再次触发写事件。我们来看函数执行的步骤：

```
readQueryFromClient()
call()
feedAppendOnlyFile()
因为rfired原因退出本次循环
下一次循环
beforeSleep()-->flushAppendOnlyFile()
aeMain()--->sendReplyToClient()
```

只有执行完了flush之后才会通知客户端数据写成功了，所以如果在feed和flush之间crash，客户接会因为进程退出接受到一个fin包，也就是一个错误的返回，所以数据并没有丢，只是执行出了错。

redis crash后，重启除了利用rdb重新载入数据外，还会读取append file（redis.c 1561）加载镜像之后的数据。

激活aof，可以在redis.conf配置文件里设置：

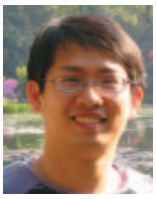
```
appendonly yes
```

也可以通过config命令在运行态启动aof：

```
cofig set appendonly yes
```

aof最大的问题就是随着时间append file会变得很大，所以我们需要bgrewriteaof命令重新整理文件，只保留最新的K-V数据，会调用rewriteAppendOnlyFile这个函数，该函数与rdbSave和类似。保存全库的K-V数据。

在作快照的过程中，K-V的变化是先写到aofbuf里。如果存在bgrewritechildpid进程，变化数据还要写到server.bgrewritebuf里（aof.c 177行）。等子进程完成快照退出之时，再由backgroundRewriteDoneHandler函数合并bgrewritebuf和全镜像两部分数据（aof.c 673行）。合并后的aof文件才是最新的全库的镜像数据。



阮若夷

目前就职于阿里云计算运维部，是阿里云关系型数据库云服务（RDS）和鹰眼监控的开发者。
博客：www.hoterran.info
微博：weibo.com/hoterran

责任编辑：高松（gaosong@csdn.net）

快速响应的增量式开发

文 / Steve Freeman, Nat Pryce

作者受到客户委托，开发一个应用程序，能够自动在拍卖中出价。他们勾画出它的工作方式和主要的组件，同时大致计划了增量式的步骤，来开发这个应用程序。

从头开始

我们是“Markup and Gouge”公司的一个开发团队，这个公司在专业市场上购买古董，再卖给“最有品位的”客户。“Markup and Gouge”一直紧随行业的发展，现在有很多购买是在网上实现的，大部分来自Southabee's，这是一家令人尊敬的拍卖行，主要发展在线业务。问题是我们的买家要花很多时间人工检查拍买的状态，并决定是否要出价，甚至有时会错失一些很有吸引力的物品，因为他们不能够快速地响应。

经过激烈的讨论，管理层决定委托开发一个“拍卖狙击者”（Auction Sniper）软件，这个应用程序将监视在线拍卖，并在价格变化时，自动以高一点的价格出价，直到达到价格上限或者拍卖结束。买家们对这个新应用程序很有兴趣，有些人同意帮助我们澄清要做的东西。

我们开始与买家小组探讨他们的想法，然后发现，要避免混淆就需要对一些基本术语达成一致意见。

- 物品是能够标识并购买的东西。
- 竞拍者是对购买物品有兴趣的人或组织。
- 竞拍出价是指一个竞拍者愿意为一件物品支付某个价格。
- 当前价格是物品的最高出价。
- 价格上限是竞拍者准备为一件物品支付的最高价格。
- 拍卖是管理一件物品竞拍出价的过程。
- 拍卖行是主持拍卖的机构。

这些讨论生成了一份很长的需求清单，例如能够针对几组相关的物品出价。没有人能够及时交付所有的需求，所以我们提出了一些选择方案，买家们不情愿地同意了，因为他们宁愿先有一个能工作的基本应用。有了这个基本应用之后，我们可以使它变得更强大。

在线系统中每件物品都有一个拍卖，所以我们决定使用物品的标识符来指代它的拍卖。在实践中，我们也发现狙击者应用不需要关心购买的物品的管理，因为其他系统会处理支付和交货。

我们决定把“拍卖狙击者”写成一个Java Swing应用。它将运行在桌面上，允许用户同时对多件物品出价。它将显示正在狙击的物品的标识符、价格上限、当前的拍卖价格和状态。买家可以通过用户界面添加新的狙击物品，并根据拍卖行的事件修改显示的值。买家们还要与易用性设计人员继续讨论，但我们已经同意大致的版本如图1所示。



图1 第一个用户界面

第一个用户界面

这显然既不完整，也不漂亮，但足够让我们可以开始工作了。

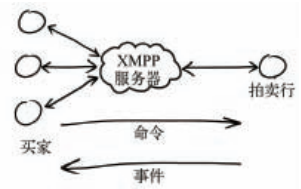


图2 Southabee's的在线拍卖系统

在进行这些讨论时，我们也与Southabee's的技术人员交谈，他们为在线服务提供支持。他们发来了描述拍卖出价协议的文档，这个协议使用XMPP（Jabber）作为底层通信层。图2展示了它处理多个竞拍者通过XMPP向拍卖行发送拍卖出价的情况，我们的狙击者应用将成为一个竞拍者。在拍卖进行时，如果某人的出价提升了当前的价格，或拍卖关闭，Southabee's将向所有连接的竞拍者发送事件，告诉他们。

与一次拍卖通信

拍卖协议

竞拍者和拍卖行之间的消息协议很简单。竞拍者发送命令，可以是：

Join（**加入**）

竞拍者加入一次拍卖。XMPP消息的发送者确定了竞拍者，聊天会话的名称确定了物品。

Bid（**竞拍出价**）

竞拍者向这次拍卖发送竞拍出价。

拍卖发送事件，可以是：

Price（**价格**）

拍卖报告当前接受的价格。这个事件也包括下次出价的最小价格增量，以及出这个价的竞拍者的名称。当竞拍者加入时，拍卖会向他发送这个事件；当新出价被接受时，拍卖会向所有竞拍者发送这个事件。

Close（**关闭**）

拍卖宣布它已关闭。最后出价者赢得此次拍卖。

我们花了一些时间来阅读文档并与Southabee's的在线支持人员交谈，整理出了一个状态机，表示了狙击者可以执行的转换。基本上，狙击者加入一次拍卖，然后经过一些轮次的出价，直到拍卖关闭，此时狙击者可能胜利或失败，参见图3。我们这时还没考虑上限价格，以使问题保持简单。

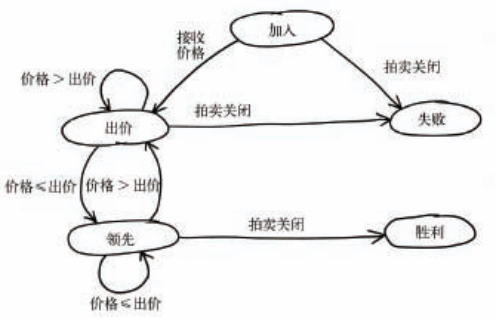


图3 竞拍者的行为可以表示为一个状态机

XMPP消息

Southabee's在线也将它们使用的XMPP消息的详细格式发送给了我们。这些格式相当简单，因为它们只涉及一些名称和值，并以键/值对的方式序列化在一行里。每一行都以协议版本号开始。这些消息看起来是这样的：

```
SOLVersion: 1.1; Command: JOIN;
SOLVersion: 1.1; Event: PRICE;
CurrentPrice: 192; Increment: 7; Bidder:
Someone else;
SOLVersion: 1.1; Command: BID; Price:
199;
SOLVersion: 1.1; Event: CLOSE;
```

Southabee's在线使用登录名称来确定要拍卖的物品，所以要竞拍标识符为12793的物品，客户端会在Southabee's服务器上与“用户”auction-12793聊天。服务器可以通过调用者的标识符来分辨谁在竞拍（假定账户都已事先建立好了）。

安全实现目标

即使是这么小一个应用也太大了，不能够一步完成，所以我们要大致弄清楚实现目标的步骤。增量式开发的关键技术就是学会将功能切片，这样就能够每次构建一点。每一小片功能应该有意义，并且足够具体，这样团队就能够断定何时它已完成。每一小片功能也应该足够小，专注于一个概念，并能够很快地实现。

将工作划分为小的、内聚的部分，这有助于管理开发的风险。我们得到定期的、具体的进展反馈，所以就可以在团队发现更多的领域知识和技术知识时，调整我们的计划。

我们眼前的任务是弄清楚狙击者应用的一系列增量式开发步骤。第一步绝对是我们能构建的最小特征。这里，骨架将以最小的路径串起Swing、XMPP和我们的应用；它仅仅是足够显示我们可以把这些组件组织在一起。后面的每一步都为已有的应用添加了一层复杂性，建立在前面工作的基础之上。经过一些讨论之后，我们得到了要构建的特征序列，如下所示：

单个物品：加入、不出价、失败

这是最开始的情况，我们将核心的基础设施放在一起。

单个物品：加入、出价、失败

在基本连接中添加出价。

单个物品：加入、出价、胜利

区分谁发出了领先的出价。

显示价格细节

开始填充用户界面。

多件物品

在同一个应用中支持竞拍多件物品。

通过用户界面添加物品

通过用户界面实现输入。

在上限价格处停止出价

使狙击者算法更智能。

在这个清单中，买家们把用户界面的优先级放在上限价格的优先级之上，部分是因为他们想确保对应用感觉舒服，还有一部分是因为如果没有用户界面，添加多件物品，并且每个带有上限价格是不容易的。

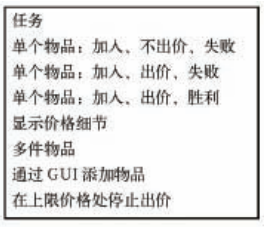


图4 初始计划

当这些功能稳定后，我们就可以处理更复杂的场景，诸如在竞价失败后重试，或使用不同的竞拍出价策略。但目前来说，实现这些功能就够我们忙的了。

虽然不知道这是否就是要采取的步骤顺序，但我们相信需要全部这些特征，并可以在做的过程中再调整。为了保持注意力，我们将计划写在一张索引卡片上，如图4所示。

这不是真的

现在您可能对我们跳过的实用性提出反对。我们之所以对过程进行了简化，目的是既让您感觉真实的项目是如何工作的，同时又保留了限制。特别是：

- 这不是一个实际的架构：XMPP既不可靠，也不安全，所以它是不适合来做交易的。确保这些品质都不在我们考虑的范围之内。也就是说，不论底层的架构是什么，我们描述的基本技术仍然适用。（再说，我们看到过主要的一些系统构建在像HTTP这样不合适的协议之上，所以这可能不像我们担心的那样不切实际。）
- 这不是敏捷计划：我们快速通过了项目计划阶段，得到了一份任务清单。在实际项目中，我们可能要了解所有的交付产物（发行计划）之后，才能动手工作。其他书籍中有关于敏捷计划的很好的介绍，如［Shore07］和［Cohn05］。
- 这不是实际可用的设计：好的用户体验设计会调查最终用户真正想做的是什么，并据此创建一致的体验。用户体验社区与敏捷开发社区已经一起工作了一段时间，探讨如何迭代地完成这项工作。这个项目足够简单，所以我们可以简要说明想要实现的目标，然后就朝着目标开始工作。



本文节选自机械工业出版社《测试驱动的面向对象软件开发》一书。该书用丰富的代码揭示TDD和OOD之间的共生关系。特此感谢机械工业出版社。

并行编程的曙光在哪？

文 / 陈冠诚

如何在多核平台上有效地编程？本文分析了并行编程的前途与发展。

加州伯克利大学的David Patterson教授在“The trouble with multicore”一文中高屋建瓴地分析了多核发展中的困难。文章开篇就说：“造芯片的家伙们正忙着生产那些大多数程序员不知道如何利用好的多核CPU。”这句话其实反映了多核CPU产生的缘由：因为功耗问题，CPU厂商没办法走继续提高单核CPU的频率以提升性能的老路，而为了利用持续增加的晶体管数量以保持性能提升，他们只好选择多核这条路。可惜的是，在他们作出这个选择时，所有程序员都不知道究竟该如何在多核平台上有效地编程。

多核的难题

那么，为什么在多核上进行并行编程很困难？如果我们把编程比作写书的话，理论上10个作者同时写一本书应该会比只有一个人写快10倍。但他们首先需要把写作任务均匀的分成10份，否则任务最多的那个作者会拖后腿，最终的写作速度肯定就快不了10倍。而光任务平均还不够，如果这个故事中的某一部分必须要在其他部分写完之后才能写，这种顺序上的依赖关系也会拖慢速度。再者，10个作者同时写一个故事，他们的故事情节必须要保持一致，那么他们肯定少不了沟通，这又拖慢了速度。这就多核并行编程时三个最大的性能挑战：负载均衡（Load Balancing）、顺序依赖关系（Sequential Dependency）和同步（Synchronization）。

难道就没有人尝试着解决这些问题吗？从20世纪60年代开始，一大批的天才们尝试着创造新的编程语言和编程模型以使得并行编程更加容易，APL、Id、Linda、Occam、SISAL等都是其中的代表。这些语言中有的确实让并行编程变容易了一些，但却没有一个能成功得像传统的串行编程语言一样兼具性能、效率和灵活性，更没有一个变得像C/C++、Java一样流行。有人会说“Java的并发包挺好用的啊，是不是足够解决多核编程的问题了呢？”其实，给编程语言加上并发库确实是一种有效的办法，但它却不能从根本上解决并行编程困难的问题。其最根本的原因就在于C/C++、Java、C#这些语言并不是天生为并发而设计的，这就决定了它们（后天添加的）并发库都只能给你提供并行编程最原始的工具，但对程序员来说并行编程却并没有因为有了这些库就变得更容易了，因为你还是得面临死锁（dead lock）、数据竞跑（data race）、伪共享（false sharing）、锁竞争（lock contention）等种种问题。

讲到这里我们就不得不提一门诞生于20多年前的古老语言Erlang了。Erlang是一门天生为并发设计的语言。它的并发模型核心是在轻量级进程间用消息传递的机制完成通信，进程之间不共享内存。这种模型的好处就在于每个进程是相互独立的，要通信就发消息好了，从

而在最大程度上减少了进程间的依赖关系。对Erlang程序来说，并发天生就能提高整体性能，而且CPU核数越多程序吞吐量（Throughput）越大，性能越好。但我们要考虑到Erlang最初是Ericsson为电信系统设计的语言，由它编写的程序的目标就是为了提高系统的吞吐量以便为更多的用户提供服务，这也是大部分服务器端程序的设计需求。这类应用的一个显著特征就是每个用户的请求在绝大部分情况下都是彼此独立的，所以多核对这样的高并发应用来讲其实有点天生一对的感觉。但对于传统的客户端程序来讲，低延迟（Latency）才是他们的首要设计目标。例如对大型的商业软件来说，它的设计目标是完成一个任务的速度更快，或者单位时间内能处理的数据更多，Erlang并不适合这样的应用。从Erlang身上我们可以看到，只有把并发融入到编程语言的设计中去，给它植入并发这个DNA才能比较好地解决并行编程的难题，现有的编程语言还需要很大的改进才能做到这点。像Scala、Go、F#这样把并发集成到设计中的新语言的出现就是一个很好的写照。

之前介绍了编程语言上的一些并行编程解决方案，那么在硬件和编程器上面有没有相关的解决方案呢？有的。另一个解决并行编程难的思路就是设计更易进行并行编程的硬件，现在学术界很火的事务性内存（Transactional Memory）就是其中的典范。但事务性内存还只处于研究阶段，里面有一大堆的问题尚待解决，其最突出的问题的就是性能还不足以满足商用需求。还有的人尝试过用编译器自动并行化，但多年的研究表明纯粹让编译器来给你进行自动并行化是完全走不通了。它能在一定程度上提升程序的性能，但非常有限，而且随着核数的增加它对性能的提升会更加有限。

那么多核时代的曙光在哪里呢？

较易并行化的应用

如果一个应用的各个子任务之间依赖关系比较小，相互独立性强，那么它就具有很好的可并行性，服务端的应用就是一个很好的例子。服务端应用的特征就是为多用户提供相似的服务，它具有内在的并行性，所以它是很适合多核的。这些应用常见的例子有大型数据库、飞机票预订系统、银行交易系统、网络搜索、游戏服务器以及云计算相关的应用等。

另一种大量采用并行化的成功案例就是多媒体处理了。举个简单的例子，渲染一幅图像这个任务就充满了大量的数据级并行（Data-level Parallelism）：一幅图像是由许许多多的像素组成的，而现在的GPU都有成百个核，我们可以比较容易地做到让每个GPU核心分别负责渲染图像的一部分，从而快速地完成整个计算任务。虽然现在来讲GPU上面的编程相对比较困难，但它所能提供的性能提升确实非常可观。

还有很火的GPGPU应用（General-purpost computing on GPU），它们在科学计算领域也有不少成功案例。其实科学计算可以算是多核上的杀手级应用了，典型的例如天气预测、气候模拟等应用。为了得到更精确的结果这些应用必须处理更多的数据，而且是必须在有限的时间内算出结果，要不然你预测后天的天气但一个礼拜才给你出结果怎么行？这些大数据量的计算任务对性能的需求永远都是非常大的。而且，这些应用本身有很多数据级的并行性，再加上这个领域一般都是行业专家和软件工程师的组合，大规模的应用并行编程是很自然的事情。

随着Big Data时代的到来，以互联网大数据量处理为代表的应用也发展得如火如荼。在这个趋势中，以Hadoop为代表的MapReduce并行编程框架大大简化了并行编程的难度，使得广大程序员能非常轻松地使用多机多核组成的集群来并行计算。MapReduce的核心思想就是简化程序员进行并行编程的复杂度，而把容错、调度和任务分割等一系列工作都交给运行时系统来完成，程序员只需要实现Map和Reduce就能通过MapReduce来挖掘多个Map任务和多个Reduce任务之间的并行性以提升性能。

整个业界都在努力

为什么我们可以对多核发展持乐观态度？因为第一点，现在整个工业界、学术界都在研

究多核，研究怎样简化并行编程、怎样降低功耗、怎样持续提升性能，大量顶尖人才都在帮助普及并行计算。第二点，动机。免费午餐都结束了，想继续提升性能？你只能并行编程。不管是客户端应用也好服务器端应用也好，用户对性能的需求肯定是不会停止的。当并行编程成为持续提升性能的唯一选择时，再困难你也得去做对不对？不过大家不用特别担心，对广大的程序员来讲，一项新技术的普及本身就是需要时间的，现在来讲大量帮助程序员进行并行编程的软硬件工具都在处在发展阶段，我们有理由相信并行编程会变得更容易。

并行编程的普及教育

因为传统的计算应用一直都以串行计算为背景，所以现在大家普遍觉得并行编程很困难。但我们换个思路看看：如果从大一开始我们就教新生《并行算法》、《并行编程导论》呢？如果程序员一开始就接受的是并行编程的教育，并行编程还是困难的吗？其实我们整个世界本身就充满了并行，人可以同时听课和做笔记，同时吃饭和交流（当然，你可以把它看做是用时间片切换的方式进行的），而计算机硬件更是从很早开始就在指令级别并行地工作，为什么软件就不可以？算法导论最新的第三版专门添加了一章《多线程算法》，（该书其中一位作者Prof. Charles Leiserson创办的并行编程公司Cilk Art也已被Intel收购）。让我大胆想象一下，整本算法导论都在讨论并行算法的时代还会远吗？



陈冠诚

IBM中国研究院研究员，瑞典查尔姆斯理工大学（Chalmers）硕士，主要研究方向为并行计算、计算机体系结构及分布式系统。个人博客为并行实验室（http://parallellabs.com）。

责任编辑：高松（gaosong@csdn.net）





主持人：张银奎
《软件调试》一书作者，从事软件开发和研究10余年，对IA-32架构、操作系统内核、虚拟技术，尤其对软件调试有较深入的研究。翻译（合译）作品包括《数据挖掘原理》、《机器学习》、《人工智能：复杂问题求解的结构和策略》、《观止——微软创建NT和未来的夺命狂奔》等。

漫谈Android系统的调试模型（中）

本期介绍用于支持应用程序调试的主机端设施。

最近，《华尔街日报》的一篇文章引起了广泛的关注，文章的作者是硅谷的著名人物马克•安德生（Marc Andreessen），文章的题目是“软件在吞噬这个世界（Why Software Is Eating The World）”。安德生因开发Mosaic浏览器和创建网景公司而成名，目前是Facebook的董事会成员，也是硅谷的著名投资人和专栏作家。在这篇文章中，安德生以近期的两则新闻开头，一则是惠普打算放弃PC业务而转向更有增长潜力的软件业务，另一则是谷歌收购摩托罗拉移动。这两则新闻让很多人感觉震动，但对于安德生来说，却正符合他所观察到的大趋势，这个大趋势就是：软件在吞噬这个世界。在解释了软件大发展的部分原因后，安德生以Oops（震惊）为小标题列举了一系列惊人的事实：世界上最大的图书销售商Amazon现在已经是一个软件公司，在云计算领域叱诧风云；在视频服务方面，世界上订阅用户最多的提供商Netflix也是软件公司；在音乐方面，具有统治地位的几家公司（苹果的iTunes、Spotify和Pandora）也都是软件公司；在电信方面，世界上增长最快的公司是刚刚被微软以85亿美元收购的Skype，也是一个彻头彻尾的软件公司；此外，世界上成长最快的猎头公司LinkedIn是软件公司，世界上最大的搜索和营销平台Google是软件公司，世界上增长最快的娱乐公司是那些视频游戏制造商，它们也都是地地道道的软件公司……总之，软件正在吞噬每一个传统行业，势不可挡，每一个行业的公司都必须做好软件革命的准备。在这篇文章的400多条评注中，很多人提到了与软件密不可分的另一个名词，那就是Bug（瑕疵）。诚然，瑕疵是软件的头号大敌，软件正在吞噬这个世界，而瑕疵也几乎无时无刻不在蚕食软件。如此说来，软件的质量关系着这个世界的安危，隐藏在软件中的瑕疵就好像是一颗颗不定时炸弹，一旦触发，就可能是灾难。如何提高软件质量呢？增强软件系统的可调试性是减少瑕疵和提高软件质量的有效方法。本期继续分析Android系统的调试模型，介绍用于支持应用程序调试的主机端设施。

再看架构

上一期我们介绍了Android系统中用于支持应用程序调试的调试模型和架构，并详细介绍了设备端的主要设施和实现方法，主要设施有两个，一是Dalvik虚拟机中对JDWP协议的实现，这部分逻辑运行在Android系统的所有Java进程中，是整个调试模型的最前端；二是运行在Android设备上的ADB监护进程，它主要承担协调连接和传输数据的职责。这一期我们继续介绍主机端（Host）的实现。图1是来自Android SDK的调试模型图，图中右下方是设备端，右上方是模拟器，左侧是主机端。

图1中画出主机端的设施有三个：第一个是使用JDWP协议的调试器，比如Eclipse，它的主要职责是源代码调试，比如设置断点、单步追踪等；第二个是DDMS，全

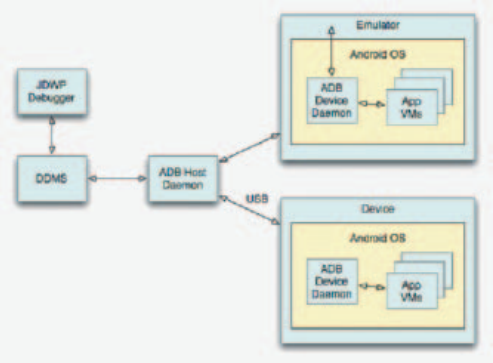


图1 架构一览

称为Dalvik Debug Monitor Server，它提供了很多种调试功能，是主机端的核心调试设施，下文会详细介绍；第三个是ADB的主机端进程，简称ADB进程，它的主要作用是与设备端的ADB监护进程通信来完成各种数据传递任务。

与Windows下的应用调试程序类似，调试Java程序也可以分为两种情况，一种情况是在调试器中创建一个新的进程并开始调试，另一种情况是将调试器附加到一个已经运行的进程。我们将分别介绍这两种情况的工作过程和原理，在此之前需要先介绍一下如何使用JDB调试Eclipse。

使用JDB调试Eclipse

JDB是JDK中包含的一个命令行界面调试器（The Java Debugger）。为了观察Eclipse调试Android程序的过程，我们将使用JDB附加到Eclipse进程，让JDB来调试Eclipse，这颇有点螳螂捕蝉，黄雀在后的意味。

JDB也是有两种方式来建立调试会话，一种方式是让JDB启动Java虚拟机和要调试的Java程序类，也就是使用类似下面这样的命令行：

```
% jdb MyClass
```

另一种方式是让JDB附加到已经运行的Java进程，这时需要使用下面这样的命令行：

```
% jdb -attach 8000
```

对于第二种方式，有一个要求，那就是要在启动Java进程时，要向Java虚拟机传递如下两个参数：

```
-Xdebug
-Xrunjdwp:transport=dt_socket,server=y,address=xxx,suspend=n
```

前一个参数的目的是启用Java虚拟机的调试支持，后一个参数的目的是指定调试参数和加载相应的支持库。

因为Eclipse是使用自己的可执行模块（eclipse.exe）来加载Java虚拟机，所以第一种方式操作起来比较困难，我们需要使用第二种方式。如何把第二种方式所需的参数传递给Java虚拟机呢？这可以通过修改Eclipse所使用的配置文件eclipse.ini来达到这个目标。在Eclipse的程序目录（比如d:\eclipse）中找到eclipse.ini，双击打开编辑，在末尾加入上面提到的两行参数就可以了（参见图2）。

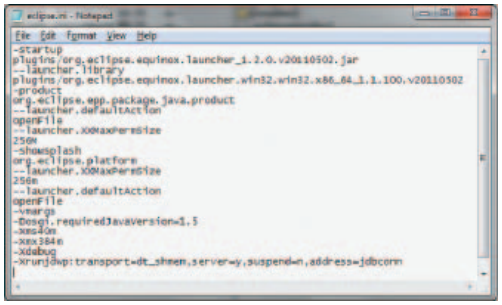


图2 配置Eclipse以便可以被JDB所调试

修改eclipse.ini后，再次启动Eclipse。为了验证效果，启动一个命令行窗口，切换到JDK的bin目录，执行如下命令：

```
jdb -attach jdbconn
```

如果成功附加，那么JDB会打印出类似图3所示的信息并显示出一个简单的命令提示符，等待我们输入调试命令。

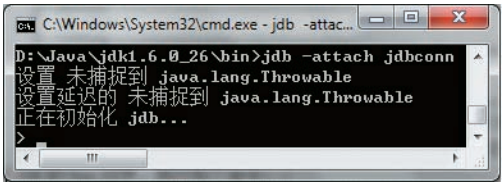


图3 将JDB附加到Eclipse

输入“？”或者help并回车，可以得到一个简单的命令列表。比如suspend命令可以挂起所有进程，resume命令可以恢复挂起的线程，threads命令可以列出所有线程，thread命令可以切换当前线程，where命令可以显示栈回溯，stop命令可以用来设置断点等。

在下文中我们将给出使用JDB命令的一些例子。

创建新的进程并调试

先启动Eclipse，打开我们上期使用的HeiAnd（Hei Android）工作空间，然后在控制台窗口启动JDB，附加到Eclipse。成功附加后，执行如下命令埋伏下一批断点：

```
> stop in com.android.ide.eclipse.adt.
internal.launch.AndroidLaunchController.
launch
> stop in com.xxx.launchApp
> stop in com.xxx.launchEmulator
> stop in com.xxx.launchRemoteDebugger
> stop in com.xxx.connectRemoteDebugger
> stop in com.xxx.doInstall
```

以上命令都是一行，不要换行，第2条命令后的xxx应该替换为android.ide.eclipse.adt.internal.launch.AndroidLaunchController，使用xxx是为了节约篇幅。如果命令执行成功，那么JDB会显示类似如下的信息：

设置断点com.xxx.launchApp

为了观察完整启动程序和开始调试的完整过程，如果已经有模拟器程序运行，那么建议先将其关闭。做好以上准备工作后，选择Eclipse的或者按F11开始调试当前的Android程序，很快JDB就会显示类似图4所示的断点命中信息。

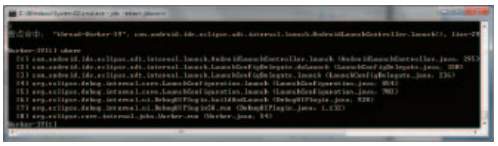


图4 开始启动被调试程序

从JDB提示的断点命中信息可以看到，被触发的断点位于com.android.ide.eclipse.adt.internal.launch.AndroidLaunchController.launch方法中。这表明位于Eclipse进程中的ADT插件正在准备启动当前的Android应用程序。执行where命令，可以看到ADT 插件被Eclipse所调用的过程。如果想从源代码角度了解更多信息，那么打开位于以下路径中的AndroidLaunchController.java源文件：

```
sdk\eclipse\plugins\com.android.ide.
eclipse.adt\src\com\android\ide\eclipse\
adt\internal\launch\
```

下面是launch方法的原型：

```
public void launch(final IProject
project,
String mode, IFile apk,
String packageName,
String debugPackageName,
Boolean debuggable,
String requiredApiVersionNumber,
final IAndroidLaunchAction launchAction,
final AndroidLaunchConfiguration config,
final AndroidLaunch launch,
IProgressMonitor monitor)
```

其中的mode参数，可以是DEBUG_MODE或者RUN_MODE，前者用来指定调试模式。Config参数用于指定启动这个程序的方式，也就是图5所示配置对话框中的信息。

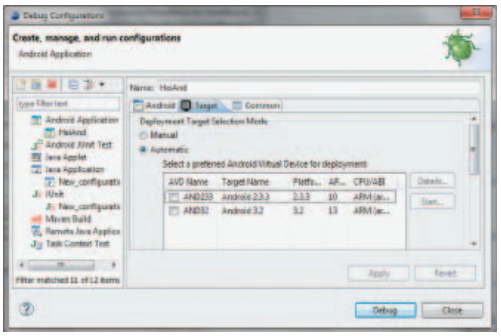


图5 配置目标设备

浏览一下launch方法的源代码，可以看到它所做的主要动作有以下几个。首先是创建一个DelayedLaunchInfo 类的实例launchInfo，然后调用它的setDebugMode方法设置是否调试，而后获取Sdk类的当前实例，从其得到Android设备（Android Virtual Device，简称AVD）管理器的实例：

```
Sdk currentSdk = Sdk.getCurrent();
AvdManager avdManager = currentSdk.
getAvdManager();
```

然后根据config参数中指定的启动模式（自动或手动）通过AVD管理器获取用来运行被调试程序的设备：

```
preferredAvd = avdManager.getAvd(config.
mAvdName, true /*validAvdOnly*/);
```

找到了合适的设备后，launch方法将执行权移交给同一个类中的continueLaunch方法让其完成后面的任务。

在JDB中执行run方法恢复Eclipse执行，很快JDB会再次显示有断点命中（篇幅关系，省略插图），这次触发断点的方法是AndroidLaunchController.launchEmulator()，也就是启动模拟器。看一下continueLaunch方法的源代码，可以看到这个方法会创建一个新的线程，这个线程会判断是否有所需的设备，如果有，便在这个设备上启动应用程序：

本期问题：

上期答案：Android的属性系统是如何实现的，简单来说，Init进程会启动一个属性服务（Property Service），这个服务会分配一段共享内存作为属性信息的缓冲区，并从.prop文件（比如/default.prop）中加载属性数据，应用程序可以调用property_get()和property_set()这两个API来读取和设置属性，这两个API内部会调用__system_property_xxx函数来通过共享内存访问属性信息。

本期问题：Android的模拟器是基于哪个著名的开源虚拟机实现的？

编者说明：

- 投稿信箱：contest@csdn.net
- 联系方式写在一个单独的TXT文件里，包括以下几项：
 1）姓名
 2）工作单位或学校
 3）电话联系方式
 4）邮寄地址
 5）E-mail地址
- 解答提交时间，最好早于当月15日。

```
launchInfo.setDevice(response.
getDeviceToUse());
simpleLaunch(launchInfo, launchInfo.
getDevice());
```

如果没有，便调用launchEmulator方法创建一个模拟器。如果想观察创建模拟器进程的细节，那么可以启动一个WinDBG，同时附加到Eclipse进程，使用如下命令设置一个断点：

```
Bp kernel32!CreateProcessA
```

设置断点后，输入g命令恢复Eclipse进程的本地代码运行，然后在JDB中执行run方法恢复Eclipse的Java代码可以运行。旋即WinDBG中的本地断点会命中，执行k命令观察栈回溯，会发现大多数栈帧都没有函数名，这是因为那些栈帧对应的函数是及时编译（JIT）产生的，WinDBG未能找到合适的方法名。

观察CreateProcessA的参数，可以看到创建的正是模拟器进程：

```
0:038> da 000000000a4a9640
00000000`0a4a9640  "D:\Android\android-
sdk\tools\emu"
00000000`0a4a9660  "lator.exe -avd
AND233 -netspeed "
00000000`0a4a9680  "full -netdelay none"
```

输入g命令恢复Eclipse执行，模拟器窗口很快跳了出来（图6）。



图6 模拟器

接下来，JDB会显示doInstall断点命中，如图7所示。从命令提示符前面的线程名称可以知道，这个断点发生在名为Monitor的工作线程中，执行where命令可以观察这个线程中正在执行的方法。从图7可以看到，安装应用程序的动作发生在Monitor线程中，被clientChanged方法所间接调用。这个方法的源代码位于AndroidLaunchController.java源文件的差不多最后面，很值得一读。

在调用syncApp做好安装（或者更新）动作后，clientChanged方法会继续调用launchApp(launchInfo, device)方法来启动要启动的程序。launchApp方法的内部会调用ActivityLaunchAction类的doLaunchAction方法执行启动动作，后者调用device.executeShellCommand方法来执行am start命令，这与在adb的shell中通过am start命令启动应用程序是等价的。在调用executeShellCommand方法时，会根据是否是调试模式来决定是否在命令行中包含-D开关：

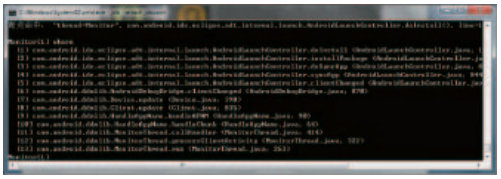


图7 安装被调试程序

```
(info.isDebugMode() ? " -D"
```

如果有-D开关，那么目标设备会等待调试器连接，显示类似图8所示的等待调用器界面。



图8 等待调试器

接下来，clientChanged方法会创建一个新的线程，并在新线程中执行connectRemoteDebugger。后者内部会使用参数中所指定的网络端口来建立连接，关键代码如下：

```
newMap.put("hostname", "localhost");
newMap.put("port", Integer.
toString(debugPort));
newMap.put("timeout", Integer.
toString(connectTimeout));
// get the default VM connector
IVMConnector connector = JavaRuntime.
getDefaultVMConnector();
// connect to remote VM
connector.connect(newMap, monitor,
launch);
```

最后两行语句特别关键，其作用是与远程的虚拟机建立JDI（Java Debug Interface）连接，连接成功后，两端就可以使用JDWP协议开始调试会话了，欲知详情，下一期将继续分析。

新书上架

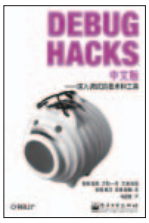
本月新书



Hadoop实战

作者：陆嘉恒

Hadoop是云计算技术的开源实现，具有高容错、跨平台等优势。本书全面讲解了HDFS和MapReduce两大核心内容，还包括Hive、HBase、Mahout、Pig、ZooKeeper、Avro、Chukwa等所有与Hadoop相关的子项目的原理及使用，以及这些子项目与Hadoop的整合使用。



DEBUG HACKS中文版

作者：吉冈弘隆 等

译者：马晶慧

本书是Miracle Linux创始人吉冈弘隆等人多年内核开发的经验积累。内容包括调试器的使用方法、基础汇编、内核错误信息捕捉、应用程序调试、内核调试。很多知识在其他领域也能派上用场，如Linux应用程序开发、嵌入式Linux开发。



写给程序员的Web设计书

作者：Hogan,B.P.

译者：吴珂

本书阐释了网站的设计与实现，帮助读者从开发人员的角度理解什么是设计。主要内容有：如何挑选配色，选择字体，用Photoshop 实现基本设计，创作Banner等页面元素，以及如何制作HTML和CSS模板以及测试设计的兼容性和可访问性等。



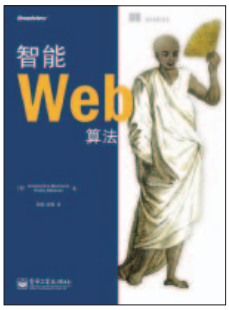
网络营销3.0

作者：Halligan,B.,Shah,D.

译者：侯德杰

HubSpot创始人Brian Halligan和Dharmesh Shah将各种形式的Web媒体联系在一起，试图打造集客式营销。本书讲解了提高Google搜索排名的工具和策略，如何利用博客来提升业绩、利用社会化媒体汇聚商业社区，以及评估在线营销成绩的各种工具。

推荐图书



智能Web算法

作者： Marmanis，H.，Babenko，D.

译者：阿稳 陈钢

出版社：电子工业出版社

本书涵盖了五类重要的智能算法：搜索、推荐、聚类、分类和分类器组合，并结合具体的案例讨论了它们在Web应用中的角色及要注意的问题。

虽然书中各章论述的是不同的理论和应用，但作者一直在试图传递一些在这个领域里不变的朴素而实用的思想，其中一个最重要的思想是：组合不同的技术，以得到更好的结果。

这无论是在学术界，还是在业界，都已经逐渐成为风气。在不久前的2011中国推荐系统峰会上，张栋博士在总结自己参加Netflix竞赛感受时就曾提到：一个好的推荐器无法打败无数技术组合起来的推荐器。而书中第3章也引述了Netflix竞赛获胜者的话：“我们没有发现完美的模型，而我们最好的结果是来自对具有互补作用的模型预测结果的组合。”

此外，书中还反复强调了一些很具有现实参考意义的观点，简要列举几点如下。

一、对问题本质及数据性质的理解比使用什么算法更重要。2011中国推荐系统峰会同样有一个类似的为人们所争议的热点，有与会者认为以重要性而论，领域知识>数据>算法。作为有过一定从业经验的算法工作者，对此应有深刻体会。

二、实验环境的数据要能代表现实数据。从统计与抽样理论的角度来说，这个命题的重要性毋庸置疑。而对于先在离线环境中训练模型，然后在生产环境中使用模型的智能应用模式而言，这点尤其重要，你是不可能把一个在北京计算得到的空气质量预测模型直接拿到北海道去用的。

三、当你想了解你的改变所产生的效果时，最好一次只改变一个因素。这其实与AB-Test的思想具有异曲同工之妙，要想研究某个参数或某个变量修改前后的差异，需要尽量保证除此之外的其他因素不变。

所以，这不单是一本介绍算法之术的书，更是一本介绍算法之道的书。读者在学习各种具体算法的同时，不妨更多地思考作者总结出来的这些实战性很强的经验，因为那些算法不一定是最新最有效的，但那些经验教训则是需要经历实际的成败方能总结出来的。此外，我们建议读者不要只看代码的实现，而更要关注蕴涵于其中的一些设计原理以及对问题建模的方式。因为在实际的应用环境中，你需要考虑的可能不仅仅是1个或2个算法，而是一个具体的系统问题。

无论是多么朴素和实用的思想，传授起来也难免让人觉得枯燥，所以作者尽其最大努力希望从工程化的角度来介绍这些算法。其中一种尝试就是尽量摒弃对数学公式的使用，而是以论述具体问题与代码说明的方式来解释算法。是否依赖数学公式来说明问题是一个值得商榷的事情，因为数学公式就好像是一种共同的语言或标准一样，只要大家都懂这种语言，沟通基本上不会存在问题，而且显得简练、优雅，无二义性。霍金也曾经说过，多一个数学公式，就会吓跑一半的读者，所以在他那本经典著作《时间简史》中只保留了最重要的一个公式。近年来，面向大众的技术书籍也有这种尽量摆脱乏味的数学公式堆砌的风潮，特别是国外的很多优秀著作，即使只用很少的公式也能把问题说清楚。

本书虽然有不少优点，但也仍有不足之处。书中在对智能技术的主要方面都有所论及，但在如此篇幅的书中把所有问题都论述清楚是不可能的，过于追求全面就会导致深度方面稍有缺乏，所有领域相关的问题未必能方方面面都讨论透彻，对于资深从业人员来说，未免会有点意

犹未尽的感觉。此外，虽然本书是本着工程的理念来讲述，在代码实现方面也考虑了一些工程上效率的因素，但毕竟只是一套教学演示代码，不免为追求简明易懂而显得简陋，所以也不能寄希望于把它们直接用于自己的生产环境中。

如作者序中所说，本书所涉及的算法是很有限的，对算法的解释也比较简要。作者的目标是选择并探讨有代表性的话题，而不是写一本代码手册或是有可能让读者晕头转向、内容包罗万象的书。



大规模Web服务开发技术
作者：伊藤直也 田中慎司
译者：李剑
出版社：机械工业出版社

正如本书的前言所说，这本书的内容主要来自Hatena株式会为学生们举行的暑期实习课程，因此其形式也是以讲义为主，风格和一般的技术读物略有不同。其内容涵盖范围也很广，从性能优化、分布式，到算法、系统架构，甚至还介绍了硬件的经济成本。最吸引人的就是书中的几个实习课题，介绍了压缩算法、全文搜索等算法的实现方法。

自己开发的Web服务，以后规模增大时系统能否承受得起？如何才能打破系统时常陷入困境的局面？面对成长起来的Web服务，整日愁眉苦脸的人一定存在吧。作者也有过完全相同的经历。

Hatena网站，月访问人次1500万，而我们参与了规模如此庞大的系统的开发和运营。1000台服务器分担负载，100多万用户不停地发表博客或收藏社会化书签，数据量每天都在激增，服务器资源频频告急。技术人员为了上GB、上TB的数据量绞尽脑汁。即便如此，流量的波涛也从未平息过。

曾几何时，Hatena的团队尚未成熟，面对规模日益增大的系统束手无策。庞大的数据和巨大的流量涌来，服务器宕机、服务停止。赶紧深夜跑过去重启服务器，觉得总算能稳定了，清晨回家之后，发现服务器又宕机了，这种事情也曾面对过。

怎样才能制服大规模服务这头野兽？本书凝聚了Hatena的技术人员在反复试验中得到的技术和诀窍——即大规模服务技术的地图和指南。

本书是开发、运营大规模服务的技术者的入门书。Web服务在不断成长，当它的数据规模变大，处理起来也就不再易如反掌了，此时，怎样做才是正确的？为了避免系统宕机，写代码时应该注意什么？设计高可扩展性（Scalability）的系统时应该记住哪些要点？这些都是本书的内容。

Hatena每年夏天都举办面向学生的、以就业体验为目的的实习活动（Hatena Summer Intern）。这些实习活动会让学生参与到Hatena的系统开发中。把开发经验很少的学生们与正式员工同样看待，让他们获得大规模系统开发的成功经验，这就是Hatena之道。那么，学生们在开发之前知道什么呢？正是历经坎坷学到的大规模服务的开发和运维的知识。

通过实习计划，Hatena的大规模服务技术的培训方法也有了体系。本书就是以实习课程为基础，试图来说明大规模服务技术。

本书的内容从操作系统和计算机的原理开始，介绍数据库的分布式方法、实用算法在系统中的应用、能支撑海量数据的搜索引擎的原理，以及统观系统全局的基础设施设计的知识等多个方面。

这是拥有1500万用户的Hatena才能传达的真实的、实践性的技术和现场感。缺乏经验的学生们在短短几天的培训中学到大规模服务开发所需的知识，在本书中得以系统化的概括。

全球排行榜

Amazon

01 Design for Hackers: Reverse Engineering Beauty
02 Cracking the Coding Interview: 150 Programming Questions and Solutions
03 Head First Java
04 Artificial Intelligence: A Modern Approach
05 Programming in Objective-C
06 JavaScript: The Good Parts
07 Introduction to Algorithms
08 Getting to Know ArcGIS Desktop
09 JavaScript: The Definitive Guide: Activate Your Web Pages
10 iOS Programming: The Big Nerd Ranch Guide

天珑书局（中国台湾）

01 前进Android Market！Google Android SDK实战演练
02 学徒模式：优秀软件开发者的养成之路
03 Google Android SDK开发范例大全
04 jQuery锦囊妙计
05 深入浅出Android系统移植与开发测试
06 大话数据结构
07 探索iPhone 4程序开发实战
08 Google！Android 3手机应用程序设计入门
09 鸟哥的Linux私房菜（基础学习篇）
10 Linux网络安全技术与实现

第二书店

01 实战Hadoop：开启通向云计算的捷径
02 Debug Hacks中文版：深入调试的技术和工具
03 松本行弘的程序世界
04 浪潮之巅
05 Java编程思想
06 智能Web算法
07 算法导论
08 Hadoop权威指南
09 大规模Web服务开发技术
10 深入理解Java虚拟机：JVM高级特性与最佳实践

GEEK产品







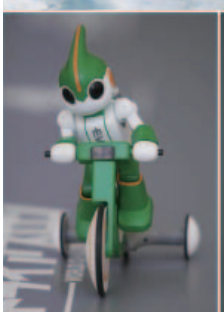
Sony HMZ-T1

Sony发布的HMZ-T1是一款人头戴式3D显示器，可以得到相当于750英寸屏幕的展示效果，显示装置采用0.7英寸高精度OLED面板（1280x720），声音系统运用虚拟环绕信号处理技术，仅通过两个耳机就可得到5.1声道音效的影院效果。

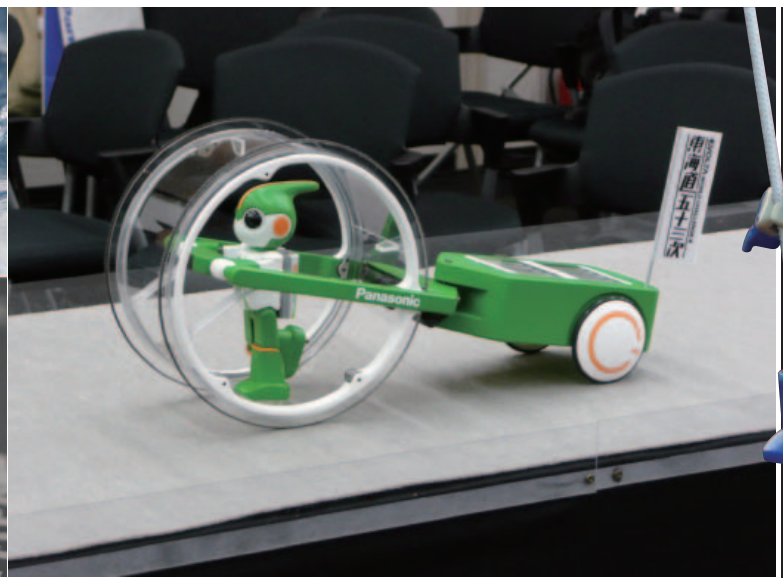
Evolta Robot

铁人三项赛可以算是挑战人类体力极限的一项考验，但对松下的这款只有手掌大小的机器人来说这可能是小菜一碟。Evolta机器人将在夏威夷挑战铁人比赛，背负三块电池以不同的身体形式完成游泳、自行车、长跑的项目。它的尺寸大概只有人的1/10，因而估计也要花费约10倍的时间来完成比赛，即大约168小时。Evolta参加这项比赛最初的目的是为了证明松下EVOLTA AA电池的优秀性能。













Dyson热力风扇

Dyson曾经因一个无扇叶的风扇而受到市场的鼓励，近日Dyson又发布了采用相同原理的冷热一体的无扇叶对流机。由于其独特的构造，风力可以送达到较远的位置，热力也可轻易地充满整个房间。该产品可以调整全方位的角度以及风力的大小。由于没有危险的加热管和扇叶，因此对于有小孩的家庭也相对安全。

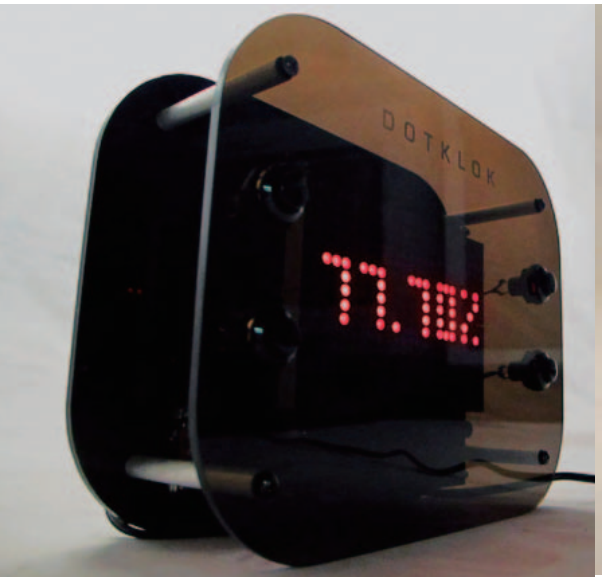


Sphero Robotic Ball

Sphero是世界上第一个球形机器人，可以通过iPhone或Android手机的蓝牙无线连接，在手机上通过触摸、点击、倾斜等方式实现对机器人的控制。开发人员也可以通过开放的API接口给Sphero创建自己的应用程序或者游戏。

DOTKLOK

DOTKLOL是一款基于开源硬件Arduino的电子时钟，且具有多种报时模式的功能。钟面屏幕由24x16颗LED灯组成，可以以“每天的秒数”、“一天的百分比”、“Ping-Pong游戏”、“数花生”、“随机模式”等各种独特的方式显示时间。









Mac OS X 背后的故事（五）

上善若水

Cordell Ratzlaff 引发的 Aqua 革命（上）

文 / 王越

Aqua是Mac OS X Public Beta全新用户界面的名字，英文中为水的词根，寓意以水为灵感，精心设计。Steve Jobs曾介绍说，Aqua的设计是如此之美好，初次见它甚至有想亲吻的冲动。本篇介绍的是Aqua的起源和来历，在下篇中，我们将展示Aqua的具体设计过程。

“Mac OS的图形界面就是你们那么业余的人设计的吗？”Steve Jobs开门见山地问。

包括Cordell Ratzlaff在内的设计师们怯怯地点头称是。“你们就是一群白痴！”Steve Jobs骂道。

这个场景发生在Steve Jobs回归不久的图形界面组组会上，前文提到的骂人的话，是他送给图形界面设计组的见面礼。【注：参见http://www.cultofmac.com/how-mac-os-x-came-to-be-exclusive-10th-anniversary-story/87889，How Mac OS X Came To Be，Leander Kahney】

不进则退的局面

Mac OS曾是图形界面设计的先驱。

从System 1开始，Mac就打破了字符终端的模式，使用图形界面和用户交互设计。但自System 1到System 7，10年过去了，界面却始终没有显著的变化。设计组一直认为，为尊重用户的习惯，定下的规矩不要轻易改动。但同时，Microsoft的变化可以说是天翻地覆，从黑屏的DOS，到全屏幕的Windows 1，再到成熟的Windows 3，最后演变到奠定当今Windows界面基础的炫丽多彩的Windows 95。用当时的眼光来看，这个变化是相当惊人的。

由于因循守旧，Mac OS在界面设计上从领先掉到了最后。旧的界面原语，一成不变的界面风格，让Mac OS的图形界面在Windows



Cordell Ratzlaff讲述Aqua经历

前显得黯然无光。【注：参见http://vimeo.com/21742166，The Gap Theory of UI Design，John Gruber】

于是，在图形界面组的组会上，Steve Jobs抨击了老Mac OS界面的各种不是——几乎所有的地方都被骂了一遍。众矢之的是各种打开窗口和文件夹的方式。在Mac OS中有至少8种打开窗口和访问文件夹的方式，如弹出菜单、下拉菜单、DragStrip、Launcher、Finder等不同的程序。

Cordell Ratzlaff作为主管，他一开始担心是不是会被Steve Jobs炒掉（传闻说Steve Jobs刚进入苹果时最爱炒人，经常会发生一些“神奇”

的情况，比如有员工和他一同进了电梯，等一同出电梯时，该员工已被炒掉）。不过批评大会进行到第20分钟时，Cordell Ratzlaff转为淡定，因为他意识到如果Steve Jobs要炒他，不用废那么多话，早就可以动手了。

其实Cardell Ratzlaff是Apple内部较早意识到小组设计不思进取的人之一。他意识到苹果有三个重要的设计问题【注：参见Designing Interactions 第二章My PC 附录访谈】。第一、Apple的很多界面语言不明确。例如，在老Mac OS中，删除文件的动作是把文件图标拖到废纸篓里，但当磁盘和光盘弹出时，居然也是把图标拖到废纸篓里。第二、老Mac OS不会对问题进行变通，如果有几个图标同时显示，窗口还容易操作，但如果有几十个图标或窗口，以相同的方式显示出来，那么在繁杂的页面中找寻所需内容，对使用者则是巨大的挑战。第三、Mac OS的界面过于古板，看上去还是停留在Windows 3.0阶段。总之，当时的Mac OS已经不能代表先进的生产力，也不能代表科技的前进方向，更不能让广大用户得到更多的利益。在Cardell Ratzlaff看来， Mac OS的界面面临不进则退的重大困局，非改不可。

Cordell Ratzlaff的试水

收购NeXT以后，Apple开始考虑如何把NeXTSTEP作业系统变为下一代的Apple操作系统，但界面设计组的倦怠又浮出水面。设计组认为，这是一个浩大的工程，所以他们决定照着Mac OS 8的样子改NeXTSTEP的代码，把NeXTSTEP改成System 8的样子。这并不困难，组里只需一个人就能完成这项任务，这人的工作极其无聊——像小孩子描红模，把新界面的样子临摹得和老界面一模一样。事实上，当Apple 释出Rhapsody和Mac OS X Server初版时，经典Mac OS的界面已经被学得惟妙惟肖了。

Cordell Ratzlaff认为这种混搭，是一个极其让苹果丢颜面的事情。所以，除了那个搞山寨的人以外，他召集其他人做新界面设计的图样。而由于NeXTSTEP具有强大的图形处理和动画能力，因此很多新的图样是在新系统上完成的。



Apple将“What's not a computer!”（看起来不是电脑的电脑）的概念应用在硬件外观上，设计出具有浪漫主义气质，半透明“果冻”式且具有艺术美感的iMac，这成了Aqua设计灵感的来源

20世纪90年代初，Apple和Microsoft的操作系统都素面朝天，色调简单，统一的矩形窗口。到1997~1998年，Apple的硬件外观设计取得重大进展：由后来成为金牌设计师的Jonathan Ive领衔，设计出具有浪漫主义气质、五彩斑斓的、半透明外壳、具有曲线美感的iMac，这个设计成为Cordell Ratzlaff和他的同事们设计的灵感，他们马上就作出了一个全新的界面图样来。【注：参见http://en.wikipedia.org/wiki/IMac_G3】

与此同时，Cordell Ratzlaff 着手解决前文提到的三个设计问题。第一、他提出了一个叫“实时状态”的概念。当用户拖动文件时，废纸保持原样，而如果拖动的是磁盘，那废纸篓的图标变成“弹出”的图标。第二、窗口的问题统一采用动画加以解决。比如窗口的最小化和还原都配有动画，告诉用户窗口的来去方向。当Dock项目有所增减时，项目长度和元素也会随之改变。第三、Mac OS一改死板面孔，呈现多彩的、小清新的图形界面，所有尖锐的直角都被打磨成圆弧，并且有像iMac外壳一样半透明的菜单。当时有评论指责Apple的设计太卡通缺乏权威感，其变化之大可见一斑。【注：参见http://www.aresluna.org/attached/files/usability/papers/onethousandsquarepixelsofcanvas.pdf，One



新界面的设计让人耳目一新，在功能上也有所突破。但当时有评论指责Apple的设计太卡通缺乏权威感

thousand square pixels of canvas On evolution of icons in graphical interfaces by Marcin Wichary 第五页】

Cocoa之父Bertrand Serlet，作为Cordell Ratzlaff的上司，对新界面很满意。但当时，他们认为这个新界面实现起来难度很大，既没有时间也没有资源把这个想法在Mac OS X中付诸实现。于是先前那位孤独的照葫芦画瓢的设计者只好继续工作。

Aqua只是个设想（PS出来的图样＋模拟出来的视频），还不是能用的代码。

Steve Jobs的怒火和Aqua的源头

几个月以后，Apple举办了一个所有开发小组参加的长达两天的汇报大会。Cordell Ratzlaff汇报的时间被排在两天的最后压轴出场。大多数工程师对这长达两天的大会报告早已疲倦，感叹Mac OS X剩下的的工作很艰巨，认为发布遥遥无期。于是，Cordell Ratzlaff报告成了整个报告会的最大笑场，所有工程师使出咆哮体来评价这个工作——“啊！！！你看这新界面多出位啊！！！有没有有没有！！！居然用的透明通道！！！还搞个实时的动画！！！你难道不知道你这些永远是天方夜谭不可能完成吗？？？我们工程师伤不起啊伤不起！！！”这个新设计就这样在所有Apple顶级工程师的鄙视下被废了。

无奈于此，只好无聊地让那位开发者继续复制全套经典Mac OS界面，而当Steve Jobs召集所有设计组负责人时，这个山寨版Mac OS的展示把Steve Jobs看得情绪激动，就发生了文章开头的那一幕。

Cordell Ratzlaff前来解释压轴报告的尴尬局面，暗示千里马常有而伯乐不常有的处境，还让Steve Jobs观摩了他的杰作。果然Steve Jobs看了这几张图例后大为惊异，拍着Cordell Ratzlaff的肩说：“很好！很强大！”然后让设计组不惜一切代价做成试验品。

在加班奋战的三周后，设计组用Macromedia Director完成了一个试验品。Steve Jobs亲自来Cordell Ratzlaff办公室视察了一下午。结果是他激动地握着Cordell Ratzlaff的手，吐露心声：“你是苹果里我见到的第一个智商是三位数字的人。”得到了Steve Jobs的支持，Apple的Mac OS X开发团队，更加紧密地围绕在以Cordell Ratzlaff为核心的界面设计概念周围，开发操作系统。

有缘千里来相会，无缘对面不相识。Steve Jobs和Cordell Ratzlaff算是相见恨晚。这样由Cordell Ratzlaff主导的新界面，在Steve Jobs的支持下，横扫一切困难，成为新版操作系统界面的最大亮点。

从这时到Steve Jobs正式在舞台上秀他的Mac OS X Public Beta，还有18个月。此时，系统界面革命的旅程已经开始，一道神秘的天光射向Infinity Loop，千古杰作Aqua就要在这里诞生，其光辉历程，我们下篇再谈。



王越

清华大学建筑学院毕业，现居美国。中国著名TeX开发者，非著名OpenFOAM开发者。

责任编辑：白羽中（baiyz@csdn.net）

苹果锻造者

Jonathan Ive

文 / 司斌



1967年，Jonathan Ive生于英国伦敦的一个中产家庭。几乎从青春期开始，Ive就着迷于设计，自己去描绘并亲自设计，从珠宝、发梳、汽车到舰艇，他对一切事物都兴趣浓厚。在拜访过一些设计咨询公司后，Ive最终决定学习看起来最为普遍的产品设计，并一发不可收拾。

1985年，Ive被纽卡斯尔艺术学校设计系录取。在设计顾问公司Roberts Weaver Group实习时，他的作品被称为“Jony式的（having Jony-ness），拥有某种能释放产品潜在情感的额外的东西。”到毕业时，Ive已经两次获得皇家艺术学会的学生设计大奖，并因此赢得Stamford公司的实习机会，但他决定改飞加州，去参观硅谷那些日益兴起的设计公司。这无疑为他日后从事IT产品设计提供了契机。

在毕业后的1989年，Ive和Clive Grinyer创立了Tangerine Design，也就是在这个公司，Ive进行了一款马桶的创意，但这个设计却加速了他的离开。Ive优秀的创意被客户批得体无完肤，在这之后，Ive毅然决然地加盟了苹果。当时，Tangerine也曾接手苹果的一些设计工作，Ive也因此多次到美国加州。但Ive的这一决定还是令人惊奇的，因为那时苹果正处于Jobs回归前的萧条，正是市场份额持续下跌的时期。

进入苹果的最初时间，Ive的工作很艰辛，但他非凡的设计能力还是很快显现出来。他设计了第一个能运行“牛顿”软件的PDA。但当他在1996年成为苹果设计主管时，接手的却是一个烂摊子。年仅29岁的Ive竭尽所能地与要求压低成本者斗争。他们搬走了深受设计师喜爱、用来模拟想象中产品的超级计算机Cray。这使得苹果产品开始看上去和其他公司的一样无趣。不过Ive仍然尽力引进一些新的有天赋的年轻设计师并保持士气。

1997年7月9日，Jobs回归苹果，大刀阔斧地将60多个产品削减到只剩4个，并在全世界奔走以寻求一个真正的超级设计明星，Ive差点在这场混乱的最初夭折。但凭借非凡的履历，Ive赢得了Jobs的信任，并意识到他想要的人就在身边。当Jobs强力推行自己的设计标准时，Ive成为坚决的贯彻者，当然也从中受益：拥有更自主的设计。

他们两人联手缔造了许多伟大的苹果产品。而从第一台Mac开始，他们决心将家用电脑重塑为有趣而不是难以亲近的产品。

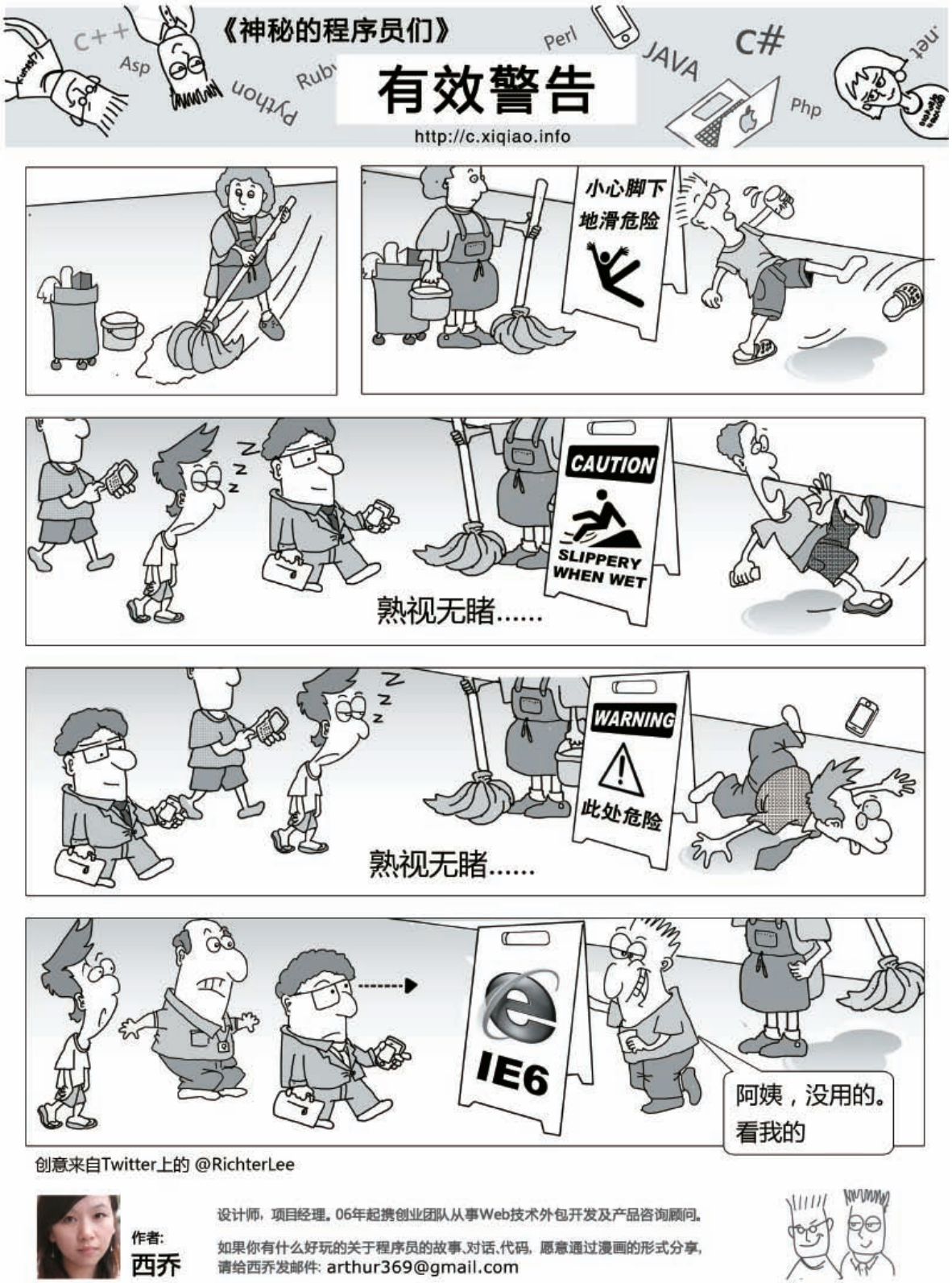
起初，Ive和其他人跑到一家糖果厂研究胶质软糖。他们和亚洲制造商一起花了几个月研究批量生产iMac的熟练工艺。设计小组甚至力争重新设计内部电子元件和线路，以保证塑料厚外壳也能看上去激动人心而不像是廉价货。但在当时，这对Jobs、Ive和苹果来说是很大的风险。

2001年，苹果推出第一台钛合金制作的电脑。幕后功臣Ive Danny De Iuliis和其他两个小组成员偷偷将价值不菲的电脑运到旧金山的仓库，在远离苹果本部的建立工作室。他们工作了6个月作出了基本设计，然后前往亚洲协商宽屏平板的事，并与精密元件生产商一起工作，最后呈现出一种干净简洁的后工业风格。这标志着iMac之前那种更趋于新奇异常的设计语言的终结。

同年10月，iPod面世，那种酷味十足的设计风格很快就使产品成为了音乐播放器中的标准。如果说最初的iMac设计延续了Ive在学校时期追求细致和新奇感的设计特色，那么颜色明快的iPod则是简约风格与时尚的合而为一。回溯了上世纪六七十年代的简约特色，当时最著名的“甲壳虫”乐队曾发行过“白色专辑”——封面仅有纯白底色和普通的黑色字体。

Ive和他的设计团队也出过差错，2000年推出的Apple G4方型电脑，因外壳容易破碎，iPod nano容易留下划痕的问题也遭到了消费者起诉。因为电池设计的问题，苹果曾召回了180万个iBook 和PowerBook笔记本电脑有潜在问题的电池。

有个电视广告的台词说：全世界都能看到我们的产品。能做到这一点的人并不多，Jonathan Ive无疑是其中之一。



创意来自Twitter上的 @RichterLee

作者:
西乔

设计师，项目经理。06年起携创业团队从事Web技术外包开发及产品咨询顾问。

如果你有什么好玩的关于程序员的故事、对话、代码，愿意通过漫画的形式分享，请给西乔发邮件：arthur369@gmail.com